BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

Hosain Alam
SPE
AU 2155
308-6662

Access DB# 127295

(49)

SEARCH REQUEST FORM

Scientific and Technical Information Center

Requester's Full Name: TRAN VU Examiner #: 79545 Date: 7-15-04
Art Unit: 2124 Phone Number 30 5 72.07 Serial Number: 09 604 113
Mail Box and Bldg/Room Location: 5Y18 Results Format Preferred (circle): PAPER DISK E-MAIL

If more than one search is submitted, please prioritize searches in order of need.

****

Please provide a detailed statement of the search topic, and describe as specifically as possible the subject matter to be searched. Include the elected species or structures, keywords, synonyms, acronyms, and registry numbers, and combine with the concept or utility of the invention. Define any terms that may have a special meaning. Give examples or relevant citations, authors, etc, if known. Please attach a copy of the cover sheet, pertinent claims, and abstract.

Title of Invention: ~~BYTE~~ METHOD & SYSTEM FOR BRANCH & CALL INSTRUCTION

Inventors (please provide full names): SZEWERENKO, SYIEK, CYRAN (TEXAS INSTR)

Earliest Priority Filing Date: 10/8/1999

***For Sequence Searches Only* Please include all pertinent information (parent, child, divisional, or issued patent numbers) along with the appropriate serial number.**

link-time modification of object code generated by the compiler for implementing far branch calls W/o changing ~~the code~~ or modifying compiler generated code. If a branch call to a target is determined to be too distant, then a trampoline code is created and control is passed thereon to effect the far call that would appear like a near call, and accept return from that far call into the trampoline. If a trampoline is already created, use it to effect the link to that far target branch address if not, create additional trampoline codes, incorporate them to the near code & pass control thereon to reach the far target; i.e. as many trampolines as needed, to set up the context switching so to make a far call look like a series of near calls..

****

STAFF USE ONLY

Searcher: Geoffrey St. Leger
Searcher Phone #: 308-7800
Searcher Location: 4B30
Date Searcher Picked Up: 7/15/4
Date Completed: 7/15/4
Searcher Prep & Review Time: 40
Clerical Prep Time:
Online Time: 170

Type of Search

NA Sequence (#)	
AA Sequence (#)	
Structure (#)	
Bibliographic	✓
Litigation	
Fulltext	✓
Patent Family	
Other	

Vendors and cost where applicable

STN
Dialog ✓
Questel/Orbit
Dr.Link
Lexis/Nexis
Sequence Systems
WWW/Internet
Other (specify)

PTO-1590 (8-01)

STIC Search Report
EIC 2100

STIC Database Tracking Number: 127[illegible]

TO:	From:
Tuan A. Vu **Location: 5Y18** **Art Unit : 2124** **Thursday, July 15, 2004** **Case Serial Number: 09/604113**	**Geoffrey St. Leger** **Location: EIC 2100** **PK2-4B30** **Phone: 308-7800** **geoffrey.stleger@uspto.gov**

Search Notes

Dear Examiner Vu,

Attached please find the results of your search request for application 09/604113. I searched Dialog's foreign patent files, technical databases, product announcement files and general files; along with ACM and the Internet.

Please let me know if you have any questions.

Regards,

Geoffrey St. Leger
4B30/308-7800

```
File 347:JAPIO Nov 1976-2004/Mar(Updated 040708)
         (c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200444
         (c) 2004  Thomson Derwent

Set     Items   Description
S1        422   TRAMPOLINE? ? OR (BRANCH OR IMPORT)()STUB? ?
S2       6727   (TARGET? ? OR DESTINATION OR BRANCH?? OR CALL??? OR ADDRESS
             OR ADDRESSES)(7N)(FAR OR DISTANT OR REMOT??)
S3       7501   (DISPLACE???? OR OFFSET OR OFF()SET)(7N)(BIG?? OR LARGE?? -
            OR MUCH OR GREAT???)
S4       1049   (UNCHANGED OR UNMODIFIED OR ("NOT" OR WITHOUT OR T)(2W)(CH-
            ANG??? OR MODIF???? OR MODIFICATION OR ALTER??? OR ALTERATION-
            ))(7N)(CODE? ? OR INSTRUCTION? ?)
S5    1272831   COMPIL? OR LINK??? OR ASSEMB???
S6          0   S1 AND S4 AND S5
S7         64   S1 AND S5
S8          3   S7 AND IC=G06F
S9          1   S1 AND S2:S3
S10        82   S4 AND S5
S11      1416   (OBJECT OR COMPILED)(1W)CODE
S12         6   S4(10N)S11
S13         7   S1 AND IC=G06F
S14         1   S2:S3 AND S10
S15         0   S1 AND S4
S16      1203   S2:S3 AND S5
S17       148   S16 AND IC=G06F
S18         5   S17 AND (COMPIL? OR ASSEMBL?) AND LINK???
S19        19   S8 OR S12:S14 OR S18
```

19/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

05603989 **Image available**
DIVISION COMPILING SYSTEM

PUB. NO.: 09-218789 [JP 9218789 A]
PUBLISHED: August 19, 1997 (19970819)
INVENTOR(s): UMA RIN
APPLICANT(s): OKI ELECTRIC IND CO LTD [000029] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 08-022222 [JP 9622222]
FILED: February 08, 1996 (19960208)
INTL CLASS: [6] G06F-009/45; G06F-009/06
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

ABSTRACT
PROBLEM TO BE SOLVED: To reduce wasteful code generation in compiling.

SOLUTION: Object information files 8 are given for respective compiled functions and an intermediate word obtained by converting a source code is recorded in the object information file 8 in compiling before the last time. The intermediate language is compared with the intermediate word obtained by converting a source code in compiling this time and a discrimination phase 5 discriminates a changed function. An object code is generated again in a code generation phase 6 on only the changed function. The function **without** the **change** is copied by the **code** generation phase 6 on the **object code** obtained in compiling before the last time.

19/5/2 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

016211980 **Image available**
WPI Acc No: 2004-369867/200435
XRPX Acc No: N04-296085
Compiler e.g. C language restricts optimization of large region variable such as static and external variables, based on designating restriction range of variable by user of compiler
Patent Assignee: RENESAS TECHNOLOGY KK (RENE-N)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 2004145589	A	20040520	JP 2002309113	A	20021024	200435	B

Priority Applications (No Type Date): JP 2002309113 A 20021024
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 2004145589	A		8	G06F-009/45	

Abstract (Basic): JP 2004145589 A
NOVELTY - The compiler restricts the optimization of large region variable such as static and external variables, based on designating the restriction range of the variable by user of the compiler.
DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for recording medium storing compiler program.
USE - Compiler such as C language.
ADVANTAGE - The optimization of variable is restricted, **without changing** the **object code** during compilation. Thereby, preventing irregularity of execution result in **object code** during call function.
DESCRIPTION OF DRAWING(S) - The figure shows the flowchart explaining the processing of the compiler. (Drawing includes non-English language text).
pp; 8 DwgNo 3/12
Title Terms: COMPILE; LANGUAGE; RESTRICT; OPTIMUM; REGION; VARIABLE; STATIC

; EXTERNAL; VARIABLE; BASED; DESIGNATED; RESTRICT; RANGE; VARIABLE; USER; COMPILE
Derwent Class: T01
International Patent Class (Main): G06F-009/45
File Segment: EPI

19/5/3 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

015394751 **Image available**
WPI Acc No: 2003-456892/200343
XRPX Acc No: N03-363387
Hint instructions provision method for processor of computer system, involves inserting break instruction into object code, such that break instruction causes processor to obtain and execute hint code
Patent Assignee: GUPTA R (GUPT-I); KARP A H (KARP-I)
Inventor: GUPTA R; KARP A H
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030061598	A1	20030327	US 2001963270	A	20010924	200343	B

Priority Applications (No Type Date): US 2001963270 A 20010924
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030061598	A1		10	G06F-009/44	

Abstract (Basic): US 20030061598 A1
NOVELTY - A hint code (64) is generated in response to a set of object code (60) to be executed by a processor (10). A break instruction is inserted into the object code, such that the break instruction causes the processor to obtain and execute the hint code.
DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for computer system.
USE - For providing hint instruction such as pre-fetch instruction or branch prediction to processor of computer system (claimed).
ADVANTAGE - Enables providing hint **instructions** to processor, **without altering object code instruction** sequences. Thus, the possibility of generation of errors during execution of object code by processor is prevented.
DESCRIPTION OF DRAWING(S) - The figure shows a computer system which provides hint instruction to processor.
processor (10)
object code (60)
hint code (64)
pp; 10 DwgNo 1/6
Title Terms: INSTRUCTION; PROVISION; METHOD; PROCESSOR; COMPUTER; SYSTEM; INSERT; BREAK; INSTRUCTION; OBJECT; CODE; BREAK; INSTRUCTION; CAUSE; PROCESSOR; OBTAIN; EXECUTE; CODE
Derwent Class: T01
International Patent Class (Main): G06F-009/44
File Segment: EPI

19/5/4 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

014946218 **Image available**
WPI Acc No: 2003-056733/200305
Related WPI Acc No: 2003-416634
XRPX Acc No: N03-043818
Run-time optimization system searches processor cache to determine whether pointer of instruction has mapping to corresponding optimized trace instruction
Patent Assignee: HEWLETT-PACKARD CO (HEWP)

Inventor: BENITEZ M; HSU W C
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6453411	B1	20020917	US 99252170	A	19990218	200305	B

Priority Applications (No Type Date): US 99252170 A 19990218
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6453411	B1		9	G06F-015/00	

Abstract (Basic): US 6453411 B1

NOVELTY - A hardware processor cache holds a mapping of instruction pointers to addresses of instructions of optimized trace stored in trace memory (105). A processor fetch unit retrieves an instruction of the program from memory for execution and searches the cache to determine whether the pointer of instruction has a mapping to corresponding trace instruction.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is included for run-time optimization method.

USE - For optimizing software applications at run-time.

ADVANTAGE - Since the run-time optimization system uses hardware managed trace address mapping, the complexity of backpatching can be eliminated, and reserving registers for passing arguments in **trampoline** code used in backpatching can be avoided. The requirement of software emulation for code profiling is also avoided. The optimization system reduces the cost of handling indirect branches. Handles execution profiling and transfers execution to optimized traces automatically, thereby allowing code to run at faster native speed without generating traces for relatively infrequent code path.

DESCRIPTION OF DRAWING(S) - The figure shows the version of the run-time optimization system.

Trace memory (105)

pp; 9 DwgNo 2/5

Title Terms: RUN; TIME; OPTIMUM; SYSTEM; SEARCH; PROCESSOR; CACHE; DETERMINE; POINT; INSTRUCTION; MAP; CORRESPOND; OPTIMUM; TRACE; INSTRUCTION
Derwent Class: T01
International Patent Class (Main): **G06F-015/00**
File Segment: EPI

19/5/5 **(Item 4 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

014219623 **Image available**
WPI Acc No: 2002-040321/200205
XRPX Acc No: N02-029805

Inter-module procedure call optimization for computer program, involves modifying call instruction to directly call unresolved module at determined location, once unresolved module is called during program execution

Patent Assignee: HEWLETT-PACKARD CO (HEWP)
Inventor: MATTSON J S
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6317870	B1	20011113	US 99258564	A	19990226	200205	B

Priority Applications (No Type Date): US 99258564 A 19990226
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6317870	B1		13	G06F-009/445	

Abstract (Basic): US 6317870 B1

NOVELTY - A call instruction is modified to call **import stub** , when call instruction executes call to unresolved module. A location of

unresolved module is determined at the time of unresolved module is called. A call instruction is modified to directly call unresolved module at determined location, once the unresolved module is called during execution of computer program.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Inter-module procedure call optimization system;

(b) Inter-module procedure call optimizing program product

USE - For optimizing inter-module procedure call in a computer program.

ADVANTAGE - Accomplishes code sharing, particularly inter-module function calls more efficiently, the call sites which are visited several times during program execution exhibit a performance improvement and the call sites which are never executed are not patched to reduce program execution time.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of user system.

pp; 13 DwgNo 1/7

Title Terms: INTER; MODULE; PROCEDURE; CALL; COMPUTER; PROGRAM; MODIFIED; CALL; INSTRUCTION; CALL; MODULE; DETERMINE; LOCATE; MODULE; CALL; PROGRAM ; EXECUTE

Derwent Class: T01

International Patent Class (Main): **G06F-009/445**

File Segment: EPI

19/5/6 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX

014075131 **Image available**

WPI Acc No: 2001-559344/200163

Related WPI Acc No: 1999-277751

XRPX Acc No: N01-415786

Computer system and interruption mechanism comprises a system processor for program execution and interruption handling from memory where interruptions are generated on test condition results

Patent Assignee: INST DEV EMERGING ARCHITECTURES LLC (EMER-N); HEWLETT-PACKARD CO (HEWP)

Inventor: HAYS J O; HUCK J C; MORRIS D C; ROSS J K

Number of Countries: 028 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 1132814	A1	20010912	EP 2001301508	A	20010220	200163	B
US 6505296	B2	20030107	US 97953836	A	19971013	200306	
			US 98168040	A	19981007		
			US 2000521160	A	20000308		
TW 485313	A	20020501	TW 2000118611	A	20000911	200318	

Priority Applications (No Type Date): US 2000521160 A 20000308; US 97953836 A 19971013; US 98168040 A 19981007

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 1132814	A1	E	27	G06F-009/46	
Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR					
US 6505296	B2			G06F-009/42	CIP of application US 97953836 CIP of application US 98168040
TW 485313	A			G06F-009/40	

Abstract (Basic): EP 1132814 A1

NOVELTY - The system comprises of a computer system (50) processor (52) including an instruction pointer executing programs and interruption handling from memory (58). The processor executes **trampoline** checks for test conditions generates an interruption if condition true supplying an address displacement to restart execution of the program at a given prestart point that is the sum of the displacement and value of the instruction pointer at time of condition

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is included for the method employed by the system in executing the program and handler for exceptional interrupt handling.

USE - The invention provides a system and method providing program execution and special handling for interruption handling with **trampoline** check instructions included emulating branch instructions.

ADVANTAGE - The inclusion of the special handler gives the option to restart the program at a pre-calculated address based on the **trampoline** checks and the address displacement

DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram of a general purpose computer on which embodiments of the invention can be implemented.

pp; 27 DwgNo 1/10

Title Terms: COMPUTER; SYSTEM; INTERRUPT; MECHANISM; COMPRISE; SYSTEM; PROCESSOR; PROGRAM; EXECUTE; INTERRUPT; HANDLE; MEMORY; INTERRUPT; GENERATE; TEST; CONDITION; RESULT

Derwent Class: T01

International Patent Class (Main): **G06F-009/40 ; G06F-009/42 ; G06F-009/46**

International Patent Class (Additional): **G06F-009/38**

File Segment: EPI

19/5/7 (Item 6 from file: 350)

[illegible]/5/7 **Image available**

WPI Acc No: 2001-534720/200159

XRPX Acc No: N01-396930

Native code instruction optimizing method for computer system, involves adjusting branch instructions of initial and final blocks relevant to order of hot trace and prediction results

Patent Assignee: HEWLETT-PACKARD CO (HEWP)

Inventor: BUZBEE W B; MATTSON J S; SHAH L V

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6205545	B1	20010320	US 9870585	A	19980430	200159	B

Priority Applications (No Type Date): US 9870585 A 19980430

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6205545	B1		11	G06F-009/32	

Abstract (Basic): US 6205545 B1

NOVELTY - A hot trace of branch instruction in basic blocks (12,14,17,21,27) are identified, based on native code instruction. The hot trace is translated and stored orderly in code cache (38). The branch instruction of block (27) is adjusted to branch into block (12) and branch instructions of other blocks are adjusted in the track order relevant to static prediction of each travel instruction.

DETAILED DESCRIPTION - A flag (42) for indicating a static prediction strategy is associated with each memory page comprising code cache. The translated branch instructions are adjusted to branch into **trampoline** blocks (44), when branch which is associated with branch instruction are predicted incorrectly. INDEPENDENT CLAIMS are also included for the following:

(a) Native code instruction optimizing apparatus in computer system;

(b) Code optimizing program storing medium

USE - For computer system.

ADVANTAGE - The basic blocks from each program are stored in code cache in single trace, thereby improves branch prediction accuracy.

DESCRIPTION OF DRAWING(S) - The figure shows the code cache and **trampoline** block with basic blocks.

Basic blocks (12,14,17,21,27)

Code cache (38)
Flag (42)
Trampoline blocks (44)
pp; 11 DwgNo 4/4
Title Terms: NATIVE; CODE; INSTRUCTION; METHOD; COMPUTER; SYSTEM; ADJUST; BRANCH; INSTRUCTION; INITIAL; FINAL; BLOCK; RELEVANT; ORDER; HOT; TRACE; PREDICT; RESULT
Derwent Class: T01
International Patent Class (Main): **G06F-009/32**
File Segment: EPI

19/5/8 (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

013904592 **Image available**
WPI Acc No: 2001-388805/200141
XRPX Acc No: N01-285885
Date field identifying method for software applications, involves altering each identified object code instruction using runtime fix code comprising year 2000 solution routine without altering source code
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: BALFOUR L C; CARTER W A; DYCK G A; LEE D E; MOORE B B; RAILSBACK G V; ROTRAMEL J D
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6240546	B1	20010529	US 98122560	A	19980724	200141	B

Priority Applications (No Type Date): US 98122560 A 19980724
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6240546	B1		12	G06F-009/45	

Abstract (Basic): US 6240546 B1
NOVELTY - A string value corresponds to potential date information having year information, is identified from a database. An object code operated with string is identified, to determine additional date in application field. A runtime fix code comprises year 2000 solution routine which is not derived from source code is provided, to alter the identified **object code instruction without altering** source **code** .
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:
(a) Date fields identifying system;
(b) Recording medium;
(c) Computer object code processing method
USE - For software applications used for processing insurance information, account information, inventory information, investment information, retirement information, etc.
ADVANTAGE - User can access object code for performing year 2000 procedures, without the need of accessing source code and recompilation of source code. Thus, modification of source code and data or input files is avoided.
DESCRIPTION OF DRAWING(S) - The figure shows the flowchart explaining the process of implementing year 2000 solution.
pp; 12 DwgNo 2/3
Title Terms: DATE; FIELD; IDENTIFY; METHOD; SOFTWARE; APPLY; ALTER; IDENTIFY; OBJECT; CODE; INSTRUCTION; FIX; CODE; COMPRISE; YEAR; SOLUTION; ROUTINE; ALTER; SOURCE; CODE
Derwent Class: T01
International Patent Class (Main): G06F-009/45
File Segment: EPI

19/5/9 (Item 8 from file: 350)

[illegible] *Image available*
[illegible] Acc No.: 2001-354422/200137
[illegible] WPI Acc No: 1999-444361; 2000-097663; 2001-024265; 2001-638809; 2002-121913
XRPX Acc No: N01-257495

Real time vehicle or equipment management system has processor with local primary focal node and modular software such as trusted remote activity controller for interfacing activity controllers of control equipment

Patent Assignee: KLINE & WALKER LLC (KLIN-N)
Inventor: WALKER R C
Number of Countries: 091 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 200078057	A1	20001221	WO 2000US16381	A	20000615	200137	B
AU 200057384	A	20010102	AU 200057384	A	20000615	200137	

Priority Applications (No Type Date): US 2000200872 P 20000501; US 99139759 P 19990615; US 2000176818 P 20000119
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200078057	A1	E	224	H04Q-001/00	

Designated States (National): AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW
Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TZ UG ZW

AU 200057384	A		H04Q-001/00	Based on patent WO 200078057

Abstract (Basic): WO 200078057 A1

NOVELTY - Processor connected to memory and communicating with external devices comprises a local primary focal node (PFN) and has programmable hardware and modular software or firmware **called** as trusted **remote** activity controller (TRAC). The software enables automated and remote control accountability for communication equipment as determined by industry or Government standard protocols and for interfacing activity controllers of control equipment.

DETAILED DESCRIPTION - The management system comprises sensory device for monitoring and reporting on data comprising command function results of peripheral devices and equipment. The memory is connected to sensory device and arranged in the vehicle or equipment for storing interface protocols for interfacing and communicating with peripheral devices that perform automated and remote control function. The external device connected to processor comprises electrical activating accessory, a peripheral device controlling automated remote control function using electricity, compressed air, gases, vacuum, hydraulic and fluid pressure, motors, mechanical or silicon relay, pistons, cylinders, pumps, valves, **linkage** levers, shifter forks, paws, ratchets, couplers, gearing or power transfer mechanism, cases, brake pads, disk **assemblies** , drums, clutches or interlocking drive mechanism, spined hub collars or shafts. The external device is [illegible] to process through a two-way communication system comprising [illegible] security device or routine to conduction signal with a security [illegible].

INDEPENDENT CLAIMS are also included for the following:

(a) Portable primary focal node (PFN) tracking device;

(b) Connectable system software (TRAC);

(c) Local PFN with trusted remote activity controller (TRAC);

(d) TRAC software record keeping of device serial numbers and personal indication numbers for its authorization and authentication program;

(e) Electrical seal system for detecting tampering and for providing water resistant seal protection;

(f) Universal communication interface for routing function;

(g) Memory for data processed using TRAC system;

(h) Remote control of actuators using PFN and processor;
(i) Accountability for activity controls confirmed by feedback sensor;
(j) Application specific sensing and supply data to monitoring or management system;
(k) Internet system for interactive highway;
(l) Interfacing or uplinking of remote monitor or management system;
(m) Switching;
(n) Machine messaging networks and computer networks;
(o) National registry to track and identify equipment and components;
(p) Spider eyes program and multitasking law enforcement tool to shutdown vehicle;
(q) Automated and remote controlled communication routing of wireless or land line;
(r) TRAC/FACT programming and hardware system for interconnecting internet

USE - Vehicle or equipment system used for management remote controlling robotic function to activate and control vehicle operation, remotely billing for use of vehicle, remotely operating machine, evaluating and diagnosing computer or processor malfunctions, remotely ordering materials and service personal to perform service and repairs, remote performing repairs electronically and remotely shutting down equipment, to restrict unauthorized use of equipment, to record and preserve data in acceptable legal manner, monitoring equipment for health and safety conditions affecting public such as reckless driving, driver impairment, pollution, vehicle unsafeness, recording and reporting monitoring gateway for billing user for use of highway, for accident investigation and machine accidents, recording audio and video of capture criminals incidents by activating and unattended vehicle system to report criminal events through remote control, recording audio and video of weather and traffic conditions, etc.

ADVANTAGE - Unauthorized access of vehicles and equipment can be prevented from remote place.

DESCRIPTION OF DRAWING(S) - The figure shows the PFN/TRAC system of four main areas of involvement comprising control security technology, mobile management, home management and commercial management.

pp; 224 DwgNo 1A/25

Title Terms: REAL; TIME; VEHICLE; EQUIPMENT; MANAGEMENT; SYSTEM; PROCESSOR; LOCAL; PRIMARY; FOCUS; NODE; MODULE; SOFTWARE; REMOTE; ACTIVE; CONTROL; INTERFACE; ACTIVE; CONTROL; CONTROL; EQUIPMENT

Derwent Class: T01; T05; W01; W02; W05

International Patent Class (Main): H04Q-001/00

International Patent Class (Additional): G01S-005/02; **G06F-007/04** ; **G06F-013/00** ; H04B-007/185; H04M-011/00

File Segment: EPI

19/5/10 **(Item 9 from file: 350)**

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

013843096 **Image available**

WPI Acc No: 2001-327309/200134

XRPX Acc No: N01-235388

Memory management method in dynamic translator used in computer systems, involves determining translated trace areas and trampoline **areas within chunks, to position translated traces and** trampoline **-instruction sets**

Patent Assignee: HEWLETT-PACKARD CO (HEWP)

Inventor: BUZBEE W B; MATTSON J S; SHAH L V

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6223339	B1	20010424	US 98149853	A	19980908	200134	B

Priority Applications (No Type Date): US 98149853 A 19980908

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes
US 6223339 B1 32 G06F-001/00

Abstract (Basic): US 6223339 B1

NOVELTY - Several chunks of computer memory unit and chunk length based on one or more machine-specific shortest jump distances, are determined. Translated trace areas within each of the chunks for positioning one or more translated traces, are determined. **Trampoline** areas within chunks for positioning one or more **trampoline** -instruction sets, are determined.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Computer system;

(b) Computer program product

USE - For dynamic translator of **compiler** used in computer system.

ADVANTAGE - Reduces the slowing effect of inefficient jumping by efficiently positioning the sections of the translated instructions with respect to each other and with respect to specialized instructions that redirect control flow from the translated instructions. Translates hot traces of original instructions, particularly with respect to a dynamic optimization system, because such selective optimization increases the likelihood that the time saved by translation will be greater than the time spent translating instructions. Improves the speed of dynamic translation systems by efficiently positioning translated instructions in the computer memory unit.

DESCRIPTION OF DRAWING(S) - The figure shows the functional block diagram of memory-managed dynamic translator of the computer system.

pp; 32 DwgNo 2/6

Title Terms: MEMORY; MANAGEMENT; METHOD; DYNAMIC; TRANSLATION; COMPUTER; SYSTEM; DETERMINE; TRANSLATION; TRACE; AREA; **TRAMPOLINE** ; AREA; CHUNK; POSITION; TRANSLATION; TRACE; **TRAMPOLINE** ; INSTRUCTION; SET

Derwent Class: T01

International Patent Class (Main): **G06F-001/00**

File Segment: EPI

19/5/11 (Item 10 from file: 350)

DIALOG(R)File 350:Derwent WPIX

013661694 **Image available**

WPI Acc No: 2001-145906/200115

XRPX Acc No: N01-106688

Direct-jump effecting method for memory management of computer programs, involves sending control sequentially from source address to target address via each of component jumps

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: BERNSTEIN D; CIVLIN J; HABER G; MENDELSON B; NAHSHON I

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6145125	A	20001107	US 98190166	A	19981112	200115	B

Priority Applications (No Type Date): US 98190166 A 19981112

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes
US 6145125 A 7 G06F-009/42

Abstract (Basic): US 6145125 A

NOVELTY - The control is sent directly from a source address to a target address at a specific distance greater than a maximum permitted range. During program **linkage** , the direct-jump is split into two component jumps, each of which is not greater than the maximum permitted range. The control is sent sequentially from the source address to the target address via each of the component jumps.

DETAILED DESCRIPTION - During program **linkage** , direct-jump is split into two component direct-jumps, by inserting **trampoline**

containing a branch instruction between direct code segments of executable program modules. An INDEPENDENT CLAIM is also included for the direct-jump effecting program stored in recording medium.

USE - For effecting direct-jump in executable program module to target address displaced from source address by specified distance for use during memory management of computer programs.

ADVANTAGE - Enables appropriate direct-jump for jumping distance exceeding maximum permitted range using simple technique.

DESCRIPTION OF DRAWING(S) - The figure shows the flow diagram indicating the direct-jump effecting method.

pp; 7 DwgNo 1B/3

Title Terms: DIRECT; JUMP; EFFECT; METHOD; MEMORY; MANAGEMENT; COMPUTER; PROGRAM; SEND; CONTROL; SEQUENCE; SOURCE; ADDRESS; TARGET; ADDRESS; COMPONENT; JUMP

Derwent Class: T01

International Patent Class (Main): **G06F-009/42**

File Segment: EPI

19/5/12 (Item 11 from file: 350)

[illegible] ...image available...

[illegible] No: 2000-105219/200009

[illegible] No: N00-080846

Function importer for importing remote function in client system connected to computer network

Patent Assignee: ELECTRONIC DATA SYSTEMS CORP (ELDA-N)

Inventor: COTTRILL S L

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6006278	A	19991221	US 97896724	A	19970718	200009	B

Priority Applications (No Type Date): US 97896724 A 19970718

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6006278	A		6	G06F-009/44	

Abstract (Basic): US 6006278 A

NOVELTY - The function call (26) of the application (24) of the client system (12) is identified and corresponding alias (36) is determined. The **remote** procedure **call compiler** (29) identifies the function of the server system (14) in response to the query. The function is copied from the server system in the host system in response to identified function call.

DETAILED DESCRIPTION - The alias of the client system includes dynamic **link** library. The server system also includes the alias (38) for cross reference of alias of client system to function of server system.

An INDEPENDENT CLAIM is also included for server/client system operating method.

USE - For importing remote function to client system connected to computer network.

ADVANTAGE - Enables server system and client system to operate independently. Enables client system to operate in stand alone configuration. Facilitates usage of client system in locations where network connections are not available.

DESCRIPTION OF DRAWING(S) - The figure shows block diagram of computer network.

Client system (12)
Server system (14)
Application (24)
Function call (26)
Remote procedure **call compiler** (29)
Alias (36,38)
pp; 6 DwgNo 1/2

Title Terms: FUNCTION; REMOTE; FUNCTION; CLIENT; SYSTEM; CONNECT; COMPUTER; NETWORK
Derwent Class: T01
International Patent Class (Main): **G06F-009/44**
International Patent Class (Additional): **G06F-009/45** ; **G06F-013/00**
File Segment: EPI

19/5/13 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

012928372 **Image available**
WPI Acc No: 2000-100208/200009
XRPX Acc No: N00-077450
Accessing remote computer systems using the internet by creating a search before connection to the internet
Patent Assignee: IESEARCH LTD (IESE-N)
Inventor: CARLILE M; CASSIDY V
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
GB 2339516	A	20000126	GB 997851	A	19990406	200009	B
GB 2339516	B	20000705	GB 997851	A	19990406	200035	

Priority Applications (No Type Date): GB 997851 A 19990406
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
GB 2339516	A		20	G06F-017/30	
GB 2339516	B			G06F-017/30	

Abstract (Basic): GB 2339516 A
NOVELTY - A server processes information requests from a computer system (1) by retrieving a selected portion of the **target** group of data from **remote** computers (2).
DETAILED DESCRIPTION - The apparatus initiates a domain seek function (1), using the server to retrieve an integration routine stored in local memory (2). The target address is then identified (3) and a resource locator is **compiled** to **link** the server to the target system (5). The residual code stream is generated from the extracted uninterpreted source code (10) and stacked (12) for sequential accessing (15) to extract the domain name to be compared with the locally stored domain name (16).
An INDEPENDENT CLAIMS is included for an inter-computer communications method.
USE - For accessing remote computer systems using the internet.
ADVANTAGE - The user can obtain required information without the overhead of network traffic by creating the search before connection to the internet.
DESCRIPTION OF DRAWING(S) - The drawing shows flow diagram of the system in operation.
Initiate seek function (1)
Retrieve local function (2)
Identify target address (3)
Link to server (5)
Extract source code (10)
Stack code (12)
Sequential access (15)
Compare domain name (16)
pp; 20 DwgNo 1/1
Title Terms: ACCESS; REMOTE; COMPUTER; SYSTEM; SEARCH; CONNECT
Derwent Class: T01; W01
International Patent Class (Main): **G06F-017/30**
International Patent Class (Additional): H04L-029/06
File Segment: EPI

19/5/14 (Item 13 from file: 350)

12504255 **Image available**
WPI Acc No: 1999-610485/199952
XRPX Acc No: N99-449801
Class state testing method in object oriented program
Patent Assignee: MOTOROLA INC (MOTI)
Inventor: GOSSAIN D K; RIGG D M
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5974255	A	19991026	US 93137704	A	19931018	199952	B

Priority Applications (No Type Date): US 93137704 A 19931018
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5974255	A		5	G06F-009/455	

Abstract (Basic): US 5974255 A

NOVELTY - The test and corresponding secondary C++ classes having predefined inheritance structure with a module function are created. A final and a return initial complete states of the test class are compared and are executed with expected results by implicitly calling through the secondary classes.

DETAILED DESCRIPTION - The C++ list classes, having respective hierarchy are stored in a computer memory. The inheritance structure grants the test class with access to all data and module functions. The module function consists of a set state module function, a verify state module function and a test vector. The initial states are set by implicitly calling through the secondary hierarchy through inheritance structure. The verify state function is used for executing final result.

USE - For testing class state in object oriented program.

ADVANTAGE - Since both initial and final states are compared, enables testing of complete class state of object in object oriented program. Since the inheritance structure is used for testing by having verify state module function, all levels are easily verified, thus minimizing the effort thereby allowing the resultant **object code** to be **unchanged** as a result of the testing.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart illustrating the class state testing method.

pp; 5 DwgNo 1/1

Title Terms: CLASS; STATE; TEST; METHOD; OBJECT; ORIENT; PROGRAM
Derwent Class: T01
International Patent Class (Main): G06F-009/455
File Segment: EPI

19/5/15 **(Item 14 from file: 350)**

Image available
WPI Acc No: 1999-214238/199918
XRPX Acc No: N99-157642
Configurable grid type system for arranging display objects on windows of graphical user interface GUI.
Patent Assignee: AST RES INC (ASTR-N)
Inventor: BONSER D; CRAWFORD C
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5883625	A	19990316	US 96635719	A	19960422	199918	B

Priority Applications (No Type Date): US 96635719 A 19960422
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes

US 5883625 A 8 G06F-015/00

Abstract (Basic): US 5883625 A

NOVELTY - Cells are arranged inside a container (22'') according to grid configuration selected from several grid configurations using a control panel of GUI. Identifies are used to associate each object (26) on display with cell and to display the objects.

DETAILED DESCRIPTION - Identifiers defining unique position of each object are utilized to place the object in each cell. Moreover, a logical list indicating a unique coordinate for each object is used to determine logical relationship between the object. Thereby, new grid configuration is added **without changing** either the **code** of the container or **object code** of object. An INDEPENDENT CLAIM is included for display arranging method in GUI.

USE - For arranging display objects like chairs on windows of GUI.

ADVANTAGE - As identifiers are used for displaying objects in each cell different grid styles are utilizable and/or easily changeable and new grid style are easily added. Allows user to customize container by selecting from different grid configuration. Streamlines process of configuring object for several grid styles maintaining visually attractive elements.

DESCRIPTION OF DRAWING(S) - The figure shows display of window using third grid style.

Container (22')

Object (26)

pp; 8 DwgNo 3/4

Title Terms: CONFIGURATION; GRID; TYPE; SYSTEM; ARRANGE; DISPLAY; OBJECT; WINDOW; GRAPHICAL; USER; INTERFACE

Derwent Class: T01

International Patent Class (Main): G06F-015/00

File Segment: EPI

19/5/16 (Item 15 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2004 Thomson Derwent. All rts. reserv.

012089269 **Image available**

WPI Acc No: 1998-506180/199843

Related WPI Acc No: 1998-110105; 1998-446696

XRPX Acc No: N98-394635

Program control flow redirecting method in compiler **- involves modifying program when execution of several branches are predicted after** trampoline **point jumps to predicted point**

Patent Assignee: KUCK & ASSOC INC (KUCK-N)

Inventor: ROBISON A D

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5805894	A	19980908	US 95388271	A	19950213	199843	B
			US 95490130	A	19950614		
			US 96611739	A	19960306		

Priority Applications (No Type Date): US 96611739 A 19960306; US 95388271 A 19950213; US 95490130 A 19950614

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5805894	A		22	G06F-009/45	CIP of application US 95388271
					CIP of application US 95490130
					CIP of patent US 5710927

Abstract (Basic): US 5805894 A

The method involves finding **trampoline** points and subsequent branches in program where control flow merges. The program is evaluated to generate a set of assertions. A program execution is predicted for each found **trampoline** point, by examining successive actions after the **trampoline** points. The result of each branching action is predicted using the flow analysis of assertion.

The flow analysis is extended incrementally for each side effecting action along the predicted path until an unpredictable branch is encountered. The program is modified when execution of several branches are predicted after the **trampoline** point such that program execution jumps from the **trampoline** point to predicted point and produces side effects equivalent to those that occurs due to non-occurrence of jump.

ADVANTAGE - Alters suitable programs such that flag variables are removed and replaced by equivalent branching.

Dwg.4/20

Title Terms: PROGRAM; CONTROL; FLOW; REDIRECT; METHOD; **COMPILE** ; MODIFIED; PROGRAM; EXECUTE; BRANCH; PREDICT; AFTER; **TRAMPOLINE** ; POINT; JUMP; PREDICT; POINT

Derwent Class: T01

International Patent Class (Main): **G06F-009/45**

File Segment: EPI

19/5/17 (Item 16 from file: 350)

0009311149 **Image available**

WPI Acc No: 1994-111005/199414

XRPX Acc No: N94-086959

Distributed computer system with ally appts for non-DCE to DCE connections - has reduced RPC run time system in non-DCE computer linked **to ally in DCE server which forms proper RPC DCE calls to other DCE units**

Patent Assignee: BULL HN INFORMATION SYSTEMS INC (HONE)

Inventor: CARLSON B M; FARRINGTON K M; STEIN S A; YEN C S

Number of Countries: 008 Number of Patents: 010

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 590519	A2	19940406	EP 93115348	A	19930923	199414	B
AU 9346274	A	19940331	AU 9346274	A	19930910	199418	
CA 2106891	A	19940326	CA 2106891	A	19930924	199423	
EP 590519	A3	19940518	EP 93115348	A	19930923	199524	
AU 663617	B	19951012	AU 9346274	A	19930910	199548	
US 5497463	A	19960305	US 92951069	A	19920925	199615	
EP 590519	B1	19990303	EP 93115348	A	19930923	199913	
DE 69323675	E	19990408	DE 623675	A	19930923	199920	
			EP 93115348	A	19930923		
ES 2127774	T3	19990501	EP 93115348	A	19930923	199924	
CA 2106891	C	20030218	CA 2106891	A	19930924	200327	

Priority Applications (No Type Date): US 92951069 A 19920925

Cited Patents: No-SR.Pub; 4.Jnl.Ref

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 590519	A2	E	24	G06F-009/46	
Designated States (Regional): DE ES FR GB IT					
AU 9346274	A			G06F-015/16	
CA 2106891	A			G06F-015/16	
EP 590519	A3			G06F-009/46	
AU 663617	B			G06F-015/16	Previous Publ. patent AU 9346274
US 5497463	A		86	G06F-013/00	
EP 590519	B1	E		G06F-009/46	
Designated States (Regional): DE ES FR GB IT					
DE 69323675	E			G06F-009/46	Based on patent EP 590519
ES 2127774	T3			G06F-009/46	Based on patent EP 590519
CA 2106891	C	E		G06F-015/16	

Abstract (Basic): EP 590519 A

The distributed computer system includes a non-distributed environment computing interconnected to a distributed computing environment (DCE) computer. The non-DCE computer includes a limited **remote** procedure **call** (RPC) component (10-2). The DCE computer includes an Ally component (12-10).

Client applications in the non-DCE computer are **compiled** with RPC 'stubs' to form **links** to the DCE computer. Some of these stubs provide direct **links** with the DCE services. Other stubs are present in a reduced form and **link** to the Ally component which reforms the RPC calls into proper DCE RPC calls.

ADVANTAGE - Allows non-DCE computer to operate with only limited amount of RPC code.

Dwg.1a/4

Title Terms: DISTRIBUTE; COMPUTER; SYSTEM; ALLY; APPARATUS; NON; CONNECT; REDUCE; RUN; TIME; SYSTEM; NON; COMPUTER; **LINK** ; ALLY; SERVE; FORM; PROPER; CALL; UNIT

Derwent Class: T01

International Patent Class (Main): **G06F-009/46** ; **G06F-013/00** ; G06F-015/16

International Patent Class (Additional): **G06F-013/20**

File Segment: EPI

19/5/18 (Item 17 from file: 350)

DIALOG(R)File 350:Derwent WPIX

009571562 **Image available**

WPI Acc No: 1993-265108/199334

XRPX Acc No: N93-203312

Distributed program stack for multi-processor computer system - converts single-thread program for operation on multi-processor system by determining which processor should execute each program module and establishing communication between them

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: AMIT N J; MARBERG J M; SHANI U

Number of Countries: 002 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
CA 2061117	A	19930603	CA 2061117	A	19920212	199334	B
US 5659701	A	19970819	US 91801149	A	19911202	199739	
			US 94245052	A	19940517		
CA 2061117	C	19980929	CA 2061117	A	19920212	199849	

Priority Applications (No Type Date): US 91801149 A 19911202; US 94245052 A 19940517

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
CA 2061117	A		49	G06F-015/16	
US 5659701	A		22	G06F-015/16	Cont of application US 91801149
CA 2061117	C			G06F-015/16	

Abstract (Basic): CA 2061117 A

A multi-processor computer system executes a single-thread program having a number of callable procedures in program modules. The local memory of each processor contains a program stack, the object code of each module that executes on that processor, and an agent object and data structures containing **linkage** information. In addition, the local memory contains a c-stub module for each procedure executable on a different processor, and a s-stub module for each procedure in local memory that can be called by a procedure executing on another processor.

When a procedure P1 executing on processor A wishes to call a procedure P2 which executes in processor B, it issues a local call to P2's c-stub in processor A's local memory. The P2 c-stub then invokes the agent process in processor A, which communicates with a corresponding agent process in processor B. The agent process in processor B causes a P2's s-stub in processor B to issue a local call to procedure P2. The return from a procedure follows the same path in reverse. Each processor independently maintains its own version of the program stack, with stack entries referencing the locally executable procedures, local stubs, or local agents. With each local call or return, the program stack for that processor is updated appropriately.

Because each processor independently maintains its own stack, **remote** procedure **calls** are not constrained by the past calling history of a process. A procedure P1 in processor A may call a procedure P2 in processor B, which may in turn call another procedure P3 in processor A. It is therefore possible for a conventional single-thread program to be converted to operation on a multi-processor system **without** any significant **modification** to the source **code** .

USE/ADVANTAGE - Data processing software, efficiently executes a single-thread computer program on more than one processor.

Dwg.1/12

Title Terms: DISTRIBUTE; PROGRAM; STACK; MULTI; PROCESSOR; COMPUTER; SYSTEM ; CONVERT; SINGLE; THREAD; PROGRAM; OPERATE; MULTI; PROCESSOR; SYSTEM; DETERMINE; PROCESSOR; EXECUTE; PROGRAM; MODULE; ESTABLISH; COMMUNICATE
Derwent Class: T01
International Patent Class (Main): G06F-015/16
International Patent Class (Additional): G06F-009/46
File Segment: EPI

19/5/19 (Item 18 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2004 Thomson Derwent. All rts. reserv.

008441709 **Image available**
WPI Acc No: 1990-328709/199044
XRPX Acc No: N90-251662

System for attachment of user equipments to processing unit - has user data and control bits carried on transmit and receive serial link **in slot entities, in frame period**

Patent Assignee: IBM CORP (IBMC); INT BUSINESS MACHINES CORP (IBMC)
Inventor: BADAOUI M; CALVIGNAC J; CARLE G; GARCIA C; VACHEE P
Number of Countries: 005 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 394596	A	19901031	EP 89480060	A	19890425	199044	B
JP 2302146	A	19901214	JP 90103251	A	19900420	199105	
US 5119376	A	19920602	US 90506035	A	19900406	199225	
US 5237572	A	19930817	US 90506035	A	19900406	199334	
			US 92830128	A	19920131		
EP 394596	B1	19960626	EP 89480060	A	19890425	199630	
DE 68926740	E	19960801	DE 626740	A	19890425	199636	
			EP 89480060	A	19890425		

Priority Applications (No Type Date): EP 89480060 A 19890425
Cited Patents: 2.Jnl.Ref; EP 138717; EP 48781; US 4079452; US 4760573; EP 77863
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5119376	A		20	H04J-003/12	
US 5237572	A		19	H04J-003/12	Div ex application US 90506035 Div ex patent US 5119376
EP 394596	B1	E	36	G06F-013/38	

Designated States (Regional): DE FR GB

DE 68926740	E		G06F-013/38	Based on patent EP 394596

Abstract (Basic): EP 394596 A

The user data and control bits are carried on transmit and receive serial **link** and in data and control slot entities arranged in frame of period T, comprising one entity per user. These entities are allocated to the user equipment through multiplexing circuit, **link** adapters to and connecting boxes to user equipments are connected through active remote modules which are specific to the standardized interfaces of the user equipments. **Link** adapters add to the data and control slot entities an outband slot which is used for exchanging control information, such as active **remote** module **address** and type which are stored in memory to be transmitted to the line adapter.

USE/ADVANTAGE - Interconnection system for attaching a maximum number of n of equipment users EU (DCE or DTE) to the line adapter of a

communication processing unit. The advantage of the interconnection system is that the attachment of the user equipments is simplified.
(30pp Dwg.No.1/8

```
File 348:EUROPEAN PATENTS 1978-2004/Jul W01
         (c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20040701,UT=20040624
         (c) 2004 WIPO/Univentio

Set      Items   Description
S1         268   TRAMPOLINE? ? OR (BRANCH OR IMPORT)()STUB? ?
S2       15232   (TARGET? ? OR DESTINATION OR BRANCH?? OR CALL??? OR ADDRESS
              OR ADDRESSES)(7N)(FAR OR DISTANT OR REMOT??)
S3       16964   (DISPLACE???? OR OFFSET OR OFF()SET)(7N)(BIG?? OR LARGE?? -
             OR MUCH OR GREAT???)
S4        3655   (UNCHANGED OR UNMODIFIED OR ("NOT" OR WITHOUT OR T)(2W)(CH-
             ANG??? OR MODIF???? OR MODIFICATION OR ALTER??? OR ALTERATION-
             ))(7N)(CODE? ? OR INSTRUCTION? ?)
S5      728377   COMPIL? OR LINK??? OR ASSEMB???
S6           0   S1(100N)S4(100N)S5
S7          47   S1(100N)S5
S8           4   S7 AND IC=G06F
S9           1   S1(50N)S2:S4
S10          2   S1(100N)S2:S4
S11        319   S4(50N)S5
S12       3622   (OBJECT OR COMPILED)(1W)CODE
S13         49   S4(10N)S12
S14         36   S13(100N)(S1:S3 OR S5)
S15         33   S14 AND AC=US/PR
S16         16   S15 AND AY=(1970:1999)/PR
S17         13   S14 AND PY=1970:1999
S18         17   S16:S17
```

8/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

01346961
Method and apparatus for optimizing execution of load and store instructions
Verfahren und Vorrichtung zur Optimierung der Ausfuhrung von Lade-/Speicherbefehlen
Procede et appareil d'optimisation de l'execution d'instructions
PATENT ASSIGNEE:
Institute for the Development of Emerging Architectures, L.L.C., (2746713), c/o Hewlett-Packard Company, 3000 Hannover Street, Palo Alto, California 94304, (US), (Applicant designated States: all)
INVENTOR:
Morris, Dale C., P.O. Box 774000, PMB 346, Steamboat Springs, CO 80477, (US)
Bryg, William R., 18830 Perago Way, Saratoga, CA 95070, (US)
Karp, Alan H., 837 Lima Court, Palo Alto, CA 94308, (US)
Chen, William, 1477 Yukan Drive, Sunnyvale, CA 94087, (US)
LEGAL REPRESENTATIVE:
Powell, Stephen David et al (52311), WILLIAMS POWELL 4 St Paul's Churchyard, London EC4M 8AY, (GB)
PATENT (CC, No, Kind, Date): EP 1150202 A2 011031 (Basic)
EP 1150202 A3 030205
APPLICATION (CC, No, Date): EP 2001303750 010425;
PRIORITY (CC, No, Date): US 559508 000427
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: **G06F-009/38** ; **G06F-009/30**
ABSTRACT WORD COUNT: 195
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200144	815
SPEC A	(English)	200144	11305
Total word count - document A			12120
Total word count - document B			0
Total word count - documents A + B			12120

INTERNATIONAL PATENT CLASS: **G06F-009/38** ...

... **G06F-009/30**
...SPECIFICATION or chk.a instruction to identify the corresponding recovery code and then transfer control to that recovery code with a **trampoline** mechanism, such as described in the above incorporated European Patent Application 01301508.6.
The exception handler may also use the...

...to identify the location of the recovery code. The recovery code can be based on a table created by the **compiler** which includes addresses of check instructions which were added by the **compiler** to a **compiled** source program. The recovery code executed is therefore identified by which check instruction is executed.
The present invention allows instructions...

8/3,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

01326630
Computer system and interruption mechanism
Rechnersystem und Interruptvorgang
Ordinateur et mecanisme d'interruption

PATENT ASSIGNEE:
Institute for the Development of Emerging Architectures, L.L.C., (2746712), 19447 Pruneridge Avenue, Cupertino, CA 95014, (US), (Applicant designated States: all)
INVENTOR:
Morris, Dale C., 399 Pope Street, Menlo Park, California 94025, (US)
Hays, James O., 1722 Campbell Avenue, San Jose, California 95125, (US)
Keys, Jonathan K., 19630 NE 125 Court, Woodinville, WA 98072, (US)
Kahle, Jerome C., 851 Talisman Drive, Palo Alto, California 94303, (US)
LEGAL REPRESENTATIVE:
Tehan, Robert et al (72663), Williams, Powell & Associates, 4 St Paul's Churchyard, London EC4M 8AY, (GB)
PATENT (CC, No, Kind, Date): EP 1132814 A1 010912 (Basic)
APPLICATION (CC, No, Date): EP 2001301508 010220;
PRIORITY (CC, No, Date): US 521160 000308
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: **G06F-009/46** ; **G06F-009/38**
ABSTRACT WORD COUNT: 131
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200137	934
SPEC A	(English)	200137	13981
Total word count - document A			14915
Total word count - document B			0
Total word count - documents A + B			14915

INTERNATIONAL PATENT CLASS: **G06F-009/46** ...

... **G06F-009/38**

...SPECIFICATION October 13, 1997.
The present invention generally relates to the execution of instructions in computer systems for example to a **trampoline** mechanism for effecting control flow change in a computer system to emulate a branch, such as a **trampoline** mechanism employed for recovery of an exception caused by advanced or speculatively executed instructions.
Computer systems include at least one processor and memory. The memory stores program instructions, data, and an operating system. The program instructions can include a **compiler** for **compiling** application programs. The operating system controls the processor and the memory for system operations and for executing the program instructions...program which would be caused if a normal branch instruction was used to effect control flow.
One implementation of the **trampoline** mechanism is for recovery from deferre exceptions caused by instructions that are speculatively executed or executed in advance. The method...

...problems encountered during execution of advanced or speculated instructions is described in detail below. As to the use of a **trampoline** mechanism for recovery from exceptions deferred by advanced or speculative instructions, the **compiler** stored in memory 58 generates special instructions referred to as speculative instructions which are performed early and speculatively.
The work...

...instruction, described below, is used to detect whether there are any exceptional conditions. If there are any exceptional conditions, the **trampoline** mechanism invokes a special section of the program code, tailored to redo the advanced or speculatively performed work. In this...

...One embodiment of the present invention relates to executing any type of instruction segment that has been scheduled by a **compiler** to be speculatively executed, verifying the integrity of the execution of the

instructions that were speculatively executed, and executing recovery... a check instruction to identify the corresponding recovery code and then transfer control to that recovery code with a **trampoline** mechanism, such as illustrated in the flow diagrams of Figure 2 and 3 and described above. The exception handler may...

...to identify the location of the recovery code. The recovery code can be based on a table created by the **compiler** which includes addresses of check instructions which were added by the **compiler** to a **compiled** source program. The recovery code executed is

8/3,K/3 **(Item 3 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS

00921485
Method and apparatus for dynamically sizing non-contiguous runtime stacks
Verfahren und Gerat zur dynamischen Grossenanpassung nicht-anschliessender Laufzeitstapel
Methode et appareil pour mesurer dynamiquement des piles d'execution non-contigues
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View, CA 94043, (US), (Applicant designated States: all)
INVENTOR:
Long, Dean R. E., P.O. Box 268, Boulder Creek, California 95006, (US)
Bishop, Alan G., 725 Jeffrey Avenue, Campbell, California 95008, (US)
Fresko, Nedim, 725 Jeffrey Avenue, Campbell, California 95008, (US)
LEGAL REPRESENTATIVE:
Alton, Andrew (97091), Urquhart-Dykes & Lord Tower House Merrion Way, Leeds LS2 8PA, (GB)
PATENT (CC, No, Kind, Date): EP 840210 A2 980506 (Basic)
EP 840210 A3 000517
APPLICATION (CC, No, Date): EP 97308316 971020;
PRIORITY (CC, No, Date): US 740445 961029
DESIGNATED STATES: DE; FR; GB; NL; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; RO; SI
INTERNATIONAL PATENT CLASS: **G06F-009/40** ; **G06F-009/42**
ABSTRACT WORD COUNT: 127
NOTE:
Figure number on first page: 3

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) 9819 828
SPEC A (English) 9819 5320
Total word count - document A 6148
Total word count - document B 0
Total word count - documents A + B 6148

INTERNATIONAL PATENT CLASS: **G06F-009/40** ...

... **G06F-009/42**

...SPECIFICATION the compiled function. In one embodiment, the method of the invention includes calling a stack checking function that includes the **compiled** function. A determination is made if additional memory is required for executing the **compiled** function. If no additional memory is required, then the **compiled** function is called and executed. However, if additional memory is necessary, then additional memory is allocated that is discontiguous with...

...memory stack.
In one embodiment, the step of allocating additional memory includes allocating a second memory stack chunk for the **compiled** function. In another embodiment, a buffer frame function is called that is configured

to create a transparent boundary between the first and second stack chunks. A **trampoline** function is also called to call the **compiled** function in the second stack chunk. In other embodiments, stack protection locks are engaged and released with respect to the...

...includes computer-readable media containing computer-readable program code devices for allocating additional computer memory stack space for executing a **compiled** function located in a first computer memory stack chunk. In one embodiment, the computer-readable medium comprises computer program code...to provide an efficient method of allocating memory in a computer system without the effort and expense of designing new **compilers** and recompiling existing software. Using the software, methods, and execute and to examine the current stack pointer, the stack protection...

...are used in the call to the actual function. In some embodiments, the execution of prologue code, associated with a **trampoline** function, may be required prior to the execution of the actual function, and the execution of epilogue code may be...

...CLAIMS stack space is required to execute said compiled function; and
allocating said additional computer memory stack space for executing said **compiled** function if it is determined that said additional computer memory stack space is required to execute said **compiled** function, wherein said additional computer memory stack space is not contiguous with said computer memory stack.
2. The method of...

...1, wherein said step of allocating said additional computer memory stack space includes allocating a second stack chunk for said **compiled** function.
3. The method of claim 2, including the additional steps of:
calling a buffer frame function, said buffer frame function configured to create a transparent boundary between said first stack chunk and said second stack chunk;
calling a **trampoline** function, said **trampoline** function configured to call said **compiled** function using said second stack chunk; and
executing said **compiled** function on said stack chunk.
4. The method of claim 3, further including the step of releasing a stack protection...
...to allocate said additional computer memory stack space include program code devices for allocating a second stack chunk for said **compiled** function.
10. The computer-readable medium of claim 9, wherein said program code devices are configured to cause a computer...

...frame function configured to create a transparent boundary between said first stack chunk and said second stack chunk;
calling a **trampoline** function, said **trampoline** function configured to call said **compiled** function using said second stack chunk; and
executing said **compiled** function on said stack chunk.
11. The computer-readable medium of claim 8, wherein said program code devices are configured...

...point for a long jump operation.
12. A computer systemconfigured to allocate additional computer memory stack space for executing a **compiled** function, which **compiled** function is located in a first computer memory stack chunk, said computer system comprising:
computer memory configured to store at least one **compiled** computer function for execution, said computer memory being further configured to be arranged into at least one memory stack space...

18/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00954215
Method for analyzing computer performance data
Verfahren zur Analyse von Computerleistungsdaten
Procede pour l'analyse des donnees relatives a la performance d'un ordinateur
PATENT ASSIGNEE:
DIGITAL EQUIPMENT CORPORATION, (313085), 111 Powdermill Road, Maynard, Massachusetts 01754, (US), (Applicant designated States: all)
INVENTOR:
Henzinger, Monika H., 80 La Loma Drive, Menlo Park, California, (US)
Vandevoorde, Mark T., 1159 Corral Avenue, Sunnyvale, California 94086, (US)
Weihl, William, 280 Clipper Street, San Francisco, California 94114, (US)
Sites, Richard L., 145 Campo Bello Lane, Menlo Park, California 94025, (US)
Leung, Shun-Tak Albert, 1285 Montecito Avenue, No. 24, Mountain View, California 94043, (US)
LEGAL REPRESENTATIVE:
Charig, Raymond Julian (79692), Eric Potter Clarkson, Park View House, 58 The Ropewalk, Nottingham NG1 5DD, (GB)
PATENT (CC, No, Kind, Date): EP 864980 A2 980916 (Basic)
EP 864980 A3 000405
APPLICATION (CC, No, Date): EP 98301588 980304;
PRIORITY (CC, No, Date): US 814190 970310
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-011/34; G06F-009/38; G06F-009/45
ABSTRACT WORD COUNT: 130
NOTE:
Figure number on first page: 15

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9838	992
SPEC A	(English)	9838	13552
Total word count - document A			14544
Total word count - document B			0
Total word count - documents A + B			14544

...SPECIFICATION execution of instructions of system procedures.
Therefore, it is desired to profile machine executable programs **without** having to **modify** source or **object code** files so profiled programs do not need to be recompiled or **linked** . Furthermore it is desired to profile both application and system (kernel) level programs. In addition...

18/3,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00780859
Method and apparatus for run-time memory access checking and memory leak detection of a multi-threaded program
Verfahren und Vorrichtung zur Uberwachung der Speicherzugriffe eines Vielfadenprogramms
Methode et appareil de controle d'acces a la memoire pendant l'execution d'un programme a fils multiples
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View, CA 94043, (US), (applicant designated states: DE;FR;GB;IT;NL)
INVENTOR:

Rishi, Alok, 507 Vallejo Street, El Granada, California 94018, (US)
Masamitsu, Jon A., 1873 Creek Road, Livermore, California 94550, (US)
LEGAL REPRESENTATIVE:
W.P. Thompson & Co. (101051), Coopers Building, Church Street, Liverpool L1 3AB, (GB)
PATENT (CC, No, Kind, Date): EP 729097 A1 960828 (Basic)
APPLICATION (CC, No, Date): EP 96300759 960205;
PRIORITY (CC, No, Date): US 384884 950207
DESIGNATED STATES: DE; FR; GB; IT; NL
INTERNATIONAL PATENT CLASS: G06F-011/00;
ABSTRACT WORD COUNT: 101

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	1658
SPEC A	(English)	EPAB96	9337
Total word count - document A			10995
Total word count - document B			0
Total word count - documents A + B			10995

...SPECIFICATION debugger program 307 can dynamically load libraries at run-time that were not specified at **link** time. Since such loading of libraries is done dynamically in the debugger program 307, the...

...initiated, thereby delaying the choice of the user until the actual run- time. Furthermore, by **not modifying** the target program **object code** at all and thus eliminating the need to relink the object files to produce the executable program, the approach of the present method avoids the use of extra **links** . Finally, the patches are applied to an in-memory process initiated from the existing target...

18/3,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

0041004
Method and apparatus for a fast debugger fix and continue operation
Verfahren und Einrichtung zur schnellen Fehlerbehebung und Arbeitsfortsetzung eines Debuggers
Procede et appareil pour depanner et continuer l'operation d'un debougeur
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View, CA 94043, (US), (applicant designated states: DE;FR;GB;IT;SE)
INVENTOR:
Preisler, Thomas, 14933 Tahoe Way, Morgan Hill, California 95037, (US)
Gramlich, Wayne C., 726 Henrietta Avenue, Sunnyvale, California 94086, (US)
Pelegri-Llopart, Eduardo, 1731 Fordham Way, Mountain View, California 94040, (US)
Miller, Terrence C., 169 Oak Court, Menlo Park, California 94025, (US)
LEGAL REPRESENTATIVE:
Johnson, Terence Leslie (42961), Edward Evans & Co. Chancery House 53-64 Chancery Lane, London WC2A 1SD, (GB)
PATENT (CC, No, Kind, Date): EP 699996 A1 960306 (Basic)
APPLICATION (CC, No, Date): EP 95305992 950829;
PRIORITY (CC, No, Date): US 299720 940901
DESIGNATED STATES: DE; FR; GB; IT; SE
INTERNATIONAL PATENT CLASS: G06F-011/00;
ABSTRACT WORD COUNT: 198

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	1782
SPEC A	(English)	EPAB96	9983
Total word count - document A			11765

Total word count - document B 0
Total word count - documents A + B 11765

...CLAIMS error occurred in executing said specific line, if a function containing said same line was **not modified** in said recompiled **object code** file.

4. The method for debugging a computer application program of claim 2 wherein said **compiler** is configured to globalize static variables in said file of edited source code containing edits...error occurred in executing said specific line, if a function containing said same line was **not modified** in said recompiled **object code** file; and

a second control device which can pop one or more frames from a ...

...16. The computer system for debugging a computer application program of claim 14 wherein said **compiler** is configured to globalize static variables in said file of edited source code containing edits...

18/3,K/4 (Item 4 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00936977
Distributed systems with replicated files
Verteilte Systeme mit replizierten Dateien
Systemes distribues avec des fichiers dupliques
PATENT ASSIGNEE:
AT&T Corp., (589370), 32 Avenue of the Americas, New York, NY 10013-2412, (US), (Proprietor designated states: all)
INVENTOR:
Rao, Chung-Hwa Herman, 4304 Springbrook Drive, Edison, New Jersey 08820, (US)
Skarra, Andrea H., 26 Orchard Road, Chatham, New Jersey 07928, (US)
LEGAL REPRESENTATIVE:
Watts, Christopher Malcolm Kelway, Dr. (37391), Lucent Technologies (UK) Ltd, 5 Mornington Road, Woodford Green Essex, IG8 0TU, (GB)
PATENT (CC, No, Kind, Date): EP 694839 A2 960131 (Basic)
EP 694839 A3 980204
EP 694839 B1 010829
APPLICATION (CC, No, Date): EP 95305025 950719;
PRIORITY (CC, No, Date): US 282683 940729
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/46
ABSTRACT WORD COUNT: 186
NOTE:
Figure number on first page: 7

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	423
CLAIMS B	(English)	200135	574
CLAIMS B	(German)	200135	517
CLAIMS B	(French)	200135	621
SPEC A	(English)	EPAB96	15460
SPEC B	(English)	200135	15318
Total word count - document A			15888
Total word count - document B			17030
Total word count - documents A + B			32918

...SPECIFICATION the location of the library routine in a copy of library routines 207. Since the **linking** had to be done when the executable code was produced, it was not possible for...

...routines 207.

Computer systems have now been developed in which library routines may be dynamically **linked** to user programs. In such computer systems, the **linking** is done when the process which executes a user program is loaded into the memory of the computer system prior to execution. With dynamic **linking** , it is possible **without altering** the **object code** of the user program to replace one set of library routines with another and thereby...

...behavior of the system upon which the user program is operating. A description of dynamic **linking** may be found in Shared Libraries, Sun Microsystems, Inc., Mountain View, CA, May 1988.

FIG. 3 shows how dynamic **linking** may be used to alter a system's behavior. In system 1 301, user process...

...SPECIFICATION the location of the library routine in a copy of library routines 207. Since the **linking** had to be done when the executable code was produced, it was not possible for...

...routines 207.

Computer systems have now been developed in which library routines may be dynamically **linked** to user programs. In such computer systems, the **linking** is done when the process which executes a user program is loaded into the memory of the computer system prior to execution. With dynamic **linking** , it is possible **without altering** the **object code** of the user program to replace one set of library routines with another and thereby...

...behavior of the system upon which the user program is operating. A description of dynamic **linking** may be found in Shared Libraries, Sun Microsystems, Inc., Mountain View, CA, May 1988.

FIG. 3 shows how dynamic **linking** may be used to alter a system's behavior. In system 1 301, user process...

18/3,K/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

0666326
METHOD AND APPARATUS FOR AUTOMATIC ANALYSIS OF A TARGET PROGRAM.
VERFAHREN UND VORRICHTUNG ZUR AUTOMATISCHEN ANALYSE EINES ZIELPROGRAMMS
PROCEDE ET DISPOSITIF POUR L'ANALYSE AUTOMATIQUE D'UN PROGRAMME CIBLE.
PATENT ASSIGNEE:
Thinking Software, Inc., (4443990), 5636 Stevens Creek Blvd, Suite 176, Cupertino, CA 95014, (US), (Proprietor designated states: all)
INVENTOR:
Shapiro, Benjamin V., 4 Fairfield Avenue, West Caldwell, NJ 07006, (US)
LEGAL REPRESENTATIVE:
Sigh, Erik et al (83901), Zacco Denmark A/S Hans Bekkevolds Alle 7, 2900 Hellerup, (DK)
PATENT (CC, No, Kind, Date): EP 699320 A1 960306 (Basic)
EP 699320 A1 970212
EP 699320 B1 030723
WO 94027213 941124
APPLICATION (CC, No, Date): EP 94916719 940510; WO 94US5182 940510
PRIORITY (CC, No, Date): US 59208 930510
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
INTERNATIONAL PATENT CLASS: G06F-011/36
NOTE:
No A-document published by EPO
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200330	620
CLAIMS B	(German)	200330	529
CLAIMS B	(French)	200330	664
SPEC B	(English)	200330	22850

Total word count - document A 0
Total word count - document B 24663
Total word count - documents A + B 24663

...SPECIFICATION component file and the ECA reference in the Control component is updated.
The resolution of **link** between the control and executable component through the ECA is done at a run time. Therefore the **compiling**, recompiling or interpreting is needed of only newly created or modified statements of executable component...

...some situations, when only the control structure has to be rearranged, a modification to eX **Object Code** can be done **without** any **changes** to the executable component and by simply rearranging the order of the elements within eX...

...and, therefore, bypassing the corresponding element of the Executable component;
When the resolution of the **links** between the Control and Executable eX Object code components is implemented at run time, **linking** or re-**linking** of a group of two or more target processes implemented by eX Object Code is to be done by **linking** of only control components of eX Objet Code.
Since eX Machine through its high level...

18/3,K/6 **(Item 6 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

File backup file system
Dateisicherungssystem
Systeme de sauvegarde de fichier
PATENT ASSIGNEE:
AT&T Corp., (589370), 32 Avenue of the Americas, New York, NY 10013-2412, (US), (Proprietor designated states: all)
INVENTOR:
Fowler, Glenn Stephen, 2322 Lyde Place, Scotch Plains, New Jersey 07076, (US)
Huang, Yennun, 33 Linberger Drive, Bridgewater, New Jersey 08807, (US)
Korn, David Gerard, 303 Mercer Street A107, New York, New York 10003, (US)
Rao, Chung-Hwa Herman, 4304 Springbrook Drive, Edison, New Jersey 08820, (US)
LEGAL REPRESENTATIVE:
Watts, Christopher Malcolm Kelway, Dr. et al (37391), Lucent Technologies (UK) Ltd, 5 Mornington Road, Woodford Green Essex, IG8 0TU, (GB)
PATENT (CC, No, Kind, Date): EP 629950 B1 011024 (Basic)
APPLICATION (CC, No, Date): EP 94304064 940607;
PRIORITY (CC, No, Date): US 80037 930618
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-011/14
ABSTRACT WORD COUNT: 190
NOTE:
Figure number on first page: 7

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) EPABF2 232
CLAIMS B (English) 200143 642
CLAIMS B (German) 200143 658
CLAIMS B (French) 200143 667
SPEC A (English) EPABF2 8656
SPEC B (English) 200143 8931
Total word count - document A 8889
Total word count - document B 10898

Total word count - documents A + B 19787

...SPECIFICATION the location of the library routine in a copy of library routines 207. Since the **linking** had to be done when the executable code was produced, it was not possible for...
...routines 207.
Computer systems have now been developed in which library routines may be dynamically **linked** to user programs. In such computer systems, the **linking** is done when the process which executes a user program is loaded into the memory of the computer system prior to execution. With dynamic **linking** , it is possible **without altering** the **object code** of the user program to replace one set of library routines with another and thereby...

...behavior of the system upon which the user program is operating. A description of dynamic **linking** may be found in Shared Libraries, Sun Microsystems, Inc., Mountain View, CA, May 1988.
FIG. 3 shows how dynamic **linking** may be used to alter a system's behavior. In system 1 301, user process...

...SPECIFICATION the location of the library routine in a copy of library routines 207. Since the **linking** had to be done when the executable code was produced, it was not possible for...

...routines 207.
Computer systems have now been developed in which library routines may be dynamically **linked** to user programs. In such computer systems, the **linking** is done when the process which executes a user program is loaded into the memory of the computer system prior to execution. With dynamic **linking** , it is possible **without altering** the **object code** of the user program to replace one set of library routines with another and thereby...

...behavior of the system upon which the user program is operating. A description of dynamic **linking** may be found in Shared Libraries, Sun Microsystems, Inc., Mountain View, CA, May 1988.
FIG. 3 shows how dynamic **linking** may be used to alter a system's behavior. In system 1 301, user process...

18/3,K/7 (Item 7 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00459510
Compounding preprocessor for cache
Verarbeitungsprozessor zur Verbindung von Befehlen fur einen Cache-Speicher
Processeur de pretraitement pour une antememoire capable de combiner des instructions
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (Proprietor designated states: all)
INVENTOR:
Vassiliadis, Stamatis, 717 Vestal Road, Vestal, New York 13850, (US)
Blaner, Bartholomew, Danton Drive, Newark Valley, New York 13811, (US)
LEGAL REPRESENTATIVE:
Teufel, Fritz, Dipl.-Phys. et al (11855), IBM Deutschland Informationssysteme GmbH, Patentwesen und Urheberrecht, 70548 Stuttgart , (DE)
PATENT (CC, No, Kind, Date): EP 496928 A2 920805 (Basic)
EP 496928 A3 930127
EP 496928 B1 000202
APPLICATION (CC, No, Date): EP 91105248 910403;
PRIORITY (CC, No, Date): US 642011 910115
DESIGNATED STATES: AT; CH; DE; DK; ES; FR; GB; IT; LI; NL; SE
INTERNATIONAL PATENT CLASS: G06F-009/38; G06F-009/30
ABSTRACT WORD COUNT: 195
NOTE:

Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200005	394
CLAIMS B	(German)	200005	369
CLAIMS B	(French)	200005	420
SPEC B	(English)	200005	14774

Total word count - document A 0
Total word count - document B 15957
Total word count - documents A + B 15957

...SPECIFICATION need for a software compounding facility, which permits the invention to be applied to existing **instructions without modifying** their **object code** forms and which can accommodate future codes, thereby obviating modification to **compilers** or assemblers. Next, the overhead required for storage of the compounding information is limited to...

18/3,K/8 **(Item 8 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00458604
Scalable compound instruction set machine architecture
Maschinenarchitektur fur skalaren Verbundbefehlssatz
Architecture de machine pour un jeu echelonnable d'instructions combinees
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: AT;CH;DE;DK;ES;FR;GB;IT;LI;NL;SE)
INVENTOR:
Blaner, Bartholomew, Danton Drive, Newark Valley, N.Y. 13811, (US)
Vassiliadis, Stamatis, 717 Vestal Road, Vestal, N.Y. 13850, (US)
LEGAL REPRESENTATIVE:
Schafer, Wolfgang, Dipl.-Ing. (62021), IBM Deutschland Informationssysteme GmbH Patentwesen und Urheberrecht, 70548 Stuttgart, (DE)
PATENT (CC, No, Kind, Date): EP 454985 A2 911106 (Basic)
EP 454985 A3 940330
EP 454985 B1 961218
APPLICATION (CC, No, Date): EP 91104323 910320;
PRIORITY (CC, No, Date): US 519384 900504
DESIGNATED STATES: AT; CH; DE; DK; ES; FR; GB; IT; LI; NL; SE
INTERNATIONAL PATENT CLASS: G06F-009/38; G06F-015/80;
ABSTRACT WORD COUNT: 175

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	938
CLAIMS B	(English)	EPAB96	860
CLAIMS B	(German)	EPAB96	824
CLAIMS B	(French)	EPAB96	992
SPEC A	(English)	EPABF1	9440
SPEC B	(English)	EPAB96	9565

Total word count - document A 10379
Total word count - document B 12241
Total word count - documents A + B 22620

...SPECIFICATION processing existing instructions which operates on a binary instruction stream as part of a post- **compiler** , or as part of an in-memory compounder, or as part of cache instruction compounding...

...create compound instructions composed of scalar instructions which have still retained their original contents. Compound **instructions** are

created **without changing** the **object code** of the scalar **instructions** which form the compound instruction, thereby allowing existing programs to realize a performance improvement on...

...SPECIFICATION processing existing instructions which operates on a binary instruction stream as part of a post- **compiler** , or as part of an in-memory compounder, or as part of cache instruction compounding...

...create compound instructions composed of scalar instructions which have still retained their original contents. Compound **instructions** are created **without changing** the **object code** of the scalar **instructions** which form the compound instruction, thereby allowing existing programs to realize a performance improvement on...

18/3,K/9 **(Item 9 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS

00458598
General purpose compounding technique for instruction-level processors
Verbindungstechnik fur Prozessoren, die auf Befehlsebene parallel arbeiten
Technique de combinaison d'instruction pour processeurs fonctionnant parallelement sur le niveau d'instruction
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB;IT)
INVENTOR:
Vassiliadis, Stamatis, 717 Vestal Road, Vestal, N.Y. 13850, (US)
Eickemeyer, Richard James, 2-C Jane Lacey Drive, Endicott, New York 13760, (US)
LEGAL REPRESENTATIVE:
Schafer, Wolfgang, Dipl.-Ing. (62021), IBM Deutschland Informationssysteme GmbH Patentwesen und Urheberrecht, 70548 Stuttgart, (DE)
PATENT (CC, No, Kind, Date): EP 454984 A2 911106 (Basic)
EP 454984 A3 940427
EP 454984 B1 960925
APPLICATION (CC, No, Date): EP 91104317 910320;
PRIORITY (CC, No, Date): US 519382 900504
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: G06F-009/38; G06F-009/44; G06F-009/45; G06F-009/30;
ABSTRACT WORD COUNT: 157

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	947
CLAIMS B	(English)	EPAB96	381
CLAIMS B	(German)	EPAB96	379
CLAIMS B	(French)	EPAB96	420
SPEC A	(English)	EPABF1	9706
SPEC B	(English)	EPAB96	9767
Total word count - document A			10653
Total word count - document B			10947
Total word count - documents A + B			21600

...SPECIFICATION instructions which have still retained their original contents. A related object is to create compound **instructions without changing** the **object code** of the scalar **instructions** which form the compound instruction, thereby allowing existing programs to realize a performance improvement on...

...processing of instructions which operates on a binary instruction stream as part of a post- **compiler** , or as part of an in-memory compounder, or as part of cache instruction compounding...

...SPECIFICATION instructions which have still retained their original contents. A related object is to crate compound **instructions without changing** the **object code** of the scalar **instructions** which form the compound instruction, thereby allowing existing programs to realize a performance improvement on...

...processing of instructions which operates on a binary instruction stream as part of a post- **compiler** , or as part of an in-memory compounder, or as part of cache instruction compounding...

18/3,K/10 **(Item 1 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00784138
SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR A REQUEST BATCHER IN A TRANSACTION SERVICES PATTERNS ENVIRONMENT
SYSTEME, PROCEDE ET ARTICLE MANUFACTURE POUR MODULE DE MISE EN LOTS DES REQUETES DANS UN ENVIRONNEMENT CARACTERISE PAR DES SERVICES TRANSACTIONNELS
Patent Applicant/Assignee:
ACCENTURE LLP, 1661 Page Mill Road, Palo Alto, CA 94304, US, US (Residence), US (Nationality)
Inventor(s):
BOWMAN-AMUAH Michel K, 6426 Peak Vista Circle, Colorado Springs, CO 80918 , US,
Legal Representative:
HICKMAN Paul L (agent), Oppenheimer Wolff & Donnelly, LLP, 1400 Page Mills Road, Palo Alto, CA 94304, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200116733 A2-A3 20010308 (WO 0116733)
Application: WO 2000US23885 20000831 (PCT/WO US0023885)
Priority Application: US 99387575 19990831
Designated States: AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CU CZ DE DK DZ EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Publication Language: English
Filing Language: English
Fulltext Word Count: 150393

Fulltext Availability:
Detailed Description

Detailed Description
... headings, lists, paragraphs, tables, electronic forms, in-line images (images next to text), and hypertext **links** . Enhancements to the original HTML 1.0 specification include banners, the applet tag to support...sheets contain directions for how and where layout elements such as margins, fonts, headers, and **links** are displayed in Web pages.

With CSS, authors can use prograniming scripts and objects to...

18/3,K/11 **(Item 2 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00784092 **Image available**
USAGE CHARACTERISTIC BASED SOFTWARE CUSTOMIZATION
PERSONNALISATION D'UN LOGICIEL EN FONCTION DE CARACTERISTIQUES D'UTILISATION
Patent Applicant/Assignee:

PORTER Swain W,
Inventor(s):
PORTER Swain W,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200023865 A2 20000427 (WO 0023865)
Application: WO 99US24761 19991021 (PCT/WO US9924761)
Priority Application: US 98176692 19981021
Designated States: AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW GH GM KE LS MW SD SL SZ TZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 5873
Fulltext Availability:
Detailed Description

Detailed Description
... other hand, if it was determined at step 1 1 04, the usage characteristics have **not changed** sufficiently to warrant regeneration of **object code** /executables 104, scheduler 906 simply proceeds to steps I 1 1 0 to reset the...

...scheduler 906 proceeds to steps 11 12-1114, where it invokes source generator 406 and **compiler / linker** 408 to generate replacements for object code/executables 104 without taking into consideration collected usage...

18/3,K/12 **(Item 3 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00543727 **Image available**
SYSTEM AND METHOD FOR REMOTELY ANALYZING THE EXECUTION OF COMPUTER PROGRAMS
SYSTEME ET PROCEDE D'ANALYSE A DISTANCE DE L'EXECUTION DE PROGRAMMES D'ORDINATEUR
Patent Applicant/Assignee:
MUTEK SOLUTIONS LTD,
Inventor(s):
WYGODNY Shlomo,
BARBOY Dmitry,
PROUSS Georgi,
VOROBEY Anatoly,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200007100 A1 20000210 (WO 0007100)
Application: WO 99US17251 19990729 (PCT/WO US9917251)
Priority Application: US 98126120 19980730; US 98126126 19980730
Designated States: AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ CZ DE DE DK DK EE EE ES FI FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZA ZW GH GM KE LS MW SD SL SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 23254

Fulltext Availability:
Detailed Description

Detailed Description
... that even time-dependent bugs can be reliably diagnosed. As described below, this process does **not** require any **modification** to the source or **object code** files of the client 102, and can therefore be used with a client 102 that...

...data stored in the trace log file 122 (mainly remote mode). As described below, the **assembly** level information in the trace log file is converted back to a source level format...

18/3,K/13 **(Item 4 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00498877 **Image available**
OBJECT CODE ANALYSIS AND REMEDIATION SYSTEM AND METHOD
ANALYSE D'UN CODE D'OBJET, SYSTEME CORRECTIF ET PROCEDE ASSOCIE
Patent Applicant/Assignee:
DIGITS CORP,
Inventor(s):
NAGEL Robert H,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9930229 A1 **19990617**
Application: WO 98US26087 19981209 (PCT/WO US9826087)
Priority Application: US 9769211 19971211; US 98208148 19981209
Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW GH GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 14676

Patent and Priority Information (Country, Number, Date):
Patent: ... **19990617**
Fulltext Availability:
Detailed Description
Publication Year: **1999**

Detailed Description
... patches, the defined patches in the rernediation step may also be applied to directly to **unmodified compiled** program **code , without changing** program length by inserting traps to the patch code.

When a mainframe-type computer, e...

...memory, and is stored in the computer main memory space. This load module contains the " **link** -edit" of one or more "object modules", which result from **compilation** of programs written, e.g. in COBOL or other computer languages. The **link** -edit process provides a load module, which includes both header information, as well as program...or "Bridging" quickly enables program to become usable after Year 2000, even though its source **code** has **not** yet been **modified** .

Translation into Machine **Code** is the actual **compilation** of source code into **object code** . This takes only 1% of the time, in either process.

Testing requires the same amount...by substituting digital information elements having a same length as digital information elements of the **unmodified object code** , although other embodiments of the invention provide new code added to the terminus of the...

...and the use of trap tables or the like, the modified pseudoassembly code is then **assembled** into a modified object module.

The information from the object module is preferably analyzed to...

18/3,K/14 **(Item 5 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00447022
METHOD FOR LOOKUP OF PACKAGES AND CLASSES IN JAVA, AND DEVICES MAKING USE OF THIS METHOD
PROCEDE DE CONSULTATION D'ENSEMBLES ET DE CLASSES DANS JAVA, ET DISPOSITIF UTILISANT CE PROCEDE
Patent Applicant/Assignee:
  INTERNATIONAL BUSINESS MACHINES CORPORATION,
  EIRICH Thomas,
Inventor(s):
  EIRICH Thomas,
Patent and Priority Information (Country, Number, Date):
  Patent: WO 9837486 A1 **19980827**
  Application: WO 97IB135 19970218 (PCT/WO IB9700135)
  Priority Application: WO 97IB135 19970218
Designated States: JP US AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Publication Language: English
Fulltext Word Count: 4095

Patent and Priority Information (Country, Number, Date):
  Patent: ... **19980827**
Fulltext Availability:
  Detailed Description
Publication Year: **1998**

Detailed Description
... note that the implementation of the extended class loader
  according to the present invention does **not** require any **modification** of
  existing Java application sources **code** and **compiled** Java **code** . The Java
  applets need not to be aware of the fact that a client system...

18/3,K/15 **(Item 6 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00415573
LOW-LEVEL ENDIAN-INDEPENDENT MACHINE LANGUAGE PROGRAM REPRESENTATION
REPRESENTATION D'UN PROGRAMME EN LANGAGE DE BAS NIVEAU INDEPENDAMMENT DU FORMAT D'EXTREMITE
Patent Applicant/Assignee:
  GENERAL MAGIC INC,
Inventor(s):
  HORWAT Waldemar,
Patent and Priority Information (Country, Number, Date):
  Patent: WO 9806034 A1 **19980212**
  Application: WO 97US13630 19970804 (PCT/WO US9713630)
  Priority Application: US 96357 19960805
Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH HU IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK TJ TM TR TT UA UG UZ VN YU GH KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 16166

Patent and Priority Information (Country, Number, Date):
  Patent: ... **19980212**
Fulltext Availability:
  Detailed Description
Publication Year: **1998**

Detailed Description
... code make it suitable for a wide variety architecture-independent
  distribution media applications. A program **compiled** into Intercode

...code can be executed on an extensible set of modem 32-bit...

...by an Intercode translator into native code. Almost all C and C++ programs can be **compiled** into Intercode **object code without changes**, although the Intercode **object code** format is not C-specific-other procedural languages can also be supported.

An Intercode translator...

...provides the best performance-typically only 2% slower than optimized native MIPS gcc (Gnu C **compiler**) output for a 14000-line C++ program-but requires that the entire translated program fit...

18/3,K/16 (Item 7 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00334786
METHODS AND APPARATUS FOR CONCURRENT EXECUTION OF SERIAL COMPUTING INSTRUCTIONS USING COMBINATORIAL ARCHITECTURE FOR PROGRAM PARTITIONING
PROCEDES ET SYSTEME D'EXECUTION SIMULTANEE D'INSTRUCTIONS EN SERIE GRACE A UNE ARCHITECTURE COMBINATOIRE DE SEGMENTATION DE PROGRAMMES
Patent Applicant/Assignee:
BERKOVICH Semyon,
BERKOVICH Efraim,
Inventor(s):
BERKOVICH Semyon,
BERKOVICH Efraim,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9617298 A1 **19960606**
Application: WO 95US12695 19951012 (PCT/WO US9512695)
Priority Application: US 94348097 19941125
Designated States: AL AM AT AU BB BG BR BY CA CH CN CZ DE DK EE ES FI GB GE HU IS JP KE KG KP KR KZ LK LR LT LU LV MD MG MN MW MX NO NZ PL PT RO RU SD SE SG SI SK TJ TM TT UA UG UZ VN KE MW SD SZ UG AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 6376
Patent and Priority Information (Country, Number, Date):
Patent: ... **19960606**
Fulltext Availability:
Detailed Description
Publication Year: **1996**

Detailed Description
... of the PEs interactions.

Accordingly, there has been described a multiprocessing system which can run **compiled code** developed for single processor applications **without change** on a multiprocessing architecture of the invention. In doing so, the principal problems of the...

18/3,K/17 (Item 8 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00279036 **Image available**
METHOD FOR MINIMIZING UNCERTAINTY IN COMPUTER SOFTWARE PROCESSES ALLOWING FOR AUTOMATIC IDENTIFICATION OF FAULTS LOCATIONS AND LOCATIONS FOR MODIFICATIONS DUE TO NEW SYSTEM REQUIREMENTS WITH INTRODUCTION OF AN ALTERNATIVE FORM OF THE TARGET PROCESS OBJECT CODE ALLOWING FOR LESS RECOMPILATION AND RE-LINKAGE PROCESSING
PROCEDE PERMETTANT DE MINIMISER L'INCERTITUDE DANS LES PROCEDES DE LOGICIELS INFORMATIQUES, DE LOCALISER AUTOMATIQUEMENT LES DEFAUTS ET DE DETERMINER L'ENDROIT OU LES MODIFICATIONS DOIVENT ETRE EFFECTUEES

Patent Applicant/Assignee:
SHAPIRO Benjamin V,
Inventor(s):
SHAPIRO Benjamin V,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9427213 A2 **19941124**
Application: WO 94US5182 19940510 (PCT/WO US9405182)
Priority Application: US 93208 19930510
Designated States: AT AU BB BG BR BY CA CH CN CZ DE DK ES FI GB GE HU JP KG KP KR KZ LK LU LV MD MG MN MW NL NO NZ PL PT RO RU SD SE SI SK TJ TT UA US UZ VN AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 27733

Patent and Priority Information (Country, Number, Date):
Patent: ... **19941124**
Fulltext Availability:
Detailed Description
Claims
Publication Year: **1994**

Detailed Description
... component file and the ECA reference in the Control component is updated.

The resolution of **link** between the control and executable component through the ECA is done at a run time. Therefore the **compiling** , recompiling or interpreting is needed of only newly created or modified statements of executable component...

...some situations, when only the control structure has to be rearranged, a modification to XPD **Object Code** can be done **without** any **changes** to the executable component and by simply rearranging the order of the elements within XPD...

...and, therefore, bypassing the corresponding element of the Executable component;
When the resolution of the **links** between the Control and Executable XPD Object code components is implemented at run time, **linking** or re-**linking** of a group of two or more target processes implemented by XPD Object Code is to be done by **linking** of only control components of XPD Objet Code.

Since XPD Machine through its high level...

Claim
... code by limited modifications within one or two components of the XPD Object code, having: **compiling** , recompiling or interpreting only newly created or modified statements
of executable component of XPD Object...XPD Graph element corresponding to the
newly created YPD Repository element,
modification to the XPD **Object Code** is being done **without** any **changes** to the executable component and by simply rearranging the order of the elements within XPD...

...element and, therefore, bypassing the corresponding element of the Executable component;

A method of **linking** or re- **linking** of a group of two or more target processes implemented by XPD Object Code by **linking** of only control components of XPD Objet Code, where the resolution of the **links** between the Control and Executable XPD Object code
components is implemented at run time,

```
File   8:Ei Compendex(R) 1970-2004/Jul W1
         (c) 2004 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2004/May
         (c) 2004 ProQuest Info&Learning
File  65:Inside Conferences 1993-2004/Jul W2
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2004/Jul W1
         (c) 2004 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2004/Jun W3
         (c)2004 Japan Science and Tech Corp(JST)
File   6:NTIS 1964-2004/Jul W2
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2004/Jul W1
         (c) 2004 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2004/Jul W1
         (c) 2004 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Jun
         (c) 2004 The HW Wilson Co.
File 266:FEDRIP 2004/May
         Comp & dist by NTIS, Intl Copyright All Rights Res
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
File 104:AeroBase 1999-2004/Jun
         (c) 2004 Contains copyrighted material
File  62:SPIN(R) 1975-2004/May W3
         (c) 2004 American Institute of Physics
File 239:Mathsci 1940-2004/Sep
         (c) 2004 American Mathematical Society

Set      Items   Description
S3         307   TRAMPOLINE? ? OR (BRANCH OR IMPORT)()STUB? ?
S4       13853   (TARGET? ? OR DESTINATION OR BRANCH?? OR CALL??? OR ADDRESS
              OR ADDRESSES)(7N)(FAR OR DISTANT OR REMOT??)
S5       21928   (DISPLACE???? OR OFFSET OR OFF()SET)(7N)(BIG?? OR LARGE?? -
             OR MUCH OR GREAT???)
S6        1459   (UNCHANGED OR UNMODIFIED OR ("NOT" OR WITHOUT OR T)(2W)(CH-
             ANG??? OR MODIF???? OR MODIFICATION OR ALTER??? OR ALTERATION-
             ))(7N)(CODE? ? OR INSTRUCTION? ?)
S5    1739422   COMPIL? OR LINK??? OR ASSEMB???
S6          0   S1 AND S4 AND S5
S7         22   S1 AND S5
S8          0   S2:S3 AND S7
S9         18   RD S7 (unique items)
S10         0   TRAMPOLINE()(CODE? ? OR INSTRUCTION? ?)
S11        32   (BRANCH OR IMPORT)()STUB? ?
S12        23   RD (unique items)
S13        15   S12 NOT PY=2000:2004
S14        20   AU=(SZEWERENKO, L? OR SYIEK, D? OR CYRAN, R? OR CYRAN, B? -
              OR SZEWERENKO L? OR SYIEK D? OR CYRAN R? OR CYRAN B?)
S15         5   S5 AND S14
S16         3   RD (unique items)
S17       196   S4 AND S5
S18         2   S4 AND (LINKER? ? OR LINK()TIME)
S19       106   S4 AND (COMPILER? ? OR COMPILING)
S20         8   S19 AND (OBJECT()CODE OR COMPILED(1W)CODE)
S21         7   RD (unique items)
```

9/TI/1 (Item 1 from file: 8)
DIALOG(R)File 8:(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

Title: Dynamic analysis of twisting somersault motions

9/TI/2 (Item 2 from file: 8)
DIALOG(R)File 8:(c) 2004 Elsevier Eng. Info. Inc. All rts. reserv.

Title: Simulation of aerial movement. III. The determination of the angular momentum of the human body.

9/TI/3 (Item 1 from file: 35)
DIALOG(R)File 35:(c) 2004 ProQuest Info&Learning. All rts. reserv.

LARGE-DIAMETER LARGE-RATIO HOT TAP TEES

9/TI/4 (Item 1 from file: 2)
DIALOG(R)File 2:(c) 2004 Institution of Electrical Engineers. All rts. reserv.

Title: CONS should not CONS its arguments. II. Cheney on the M.T.A

9/TI/5 (Item 1 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Movement Restoration of Somersaults Using Three Dimensional Model Matching Method from its Silhouette Image

9/TI/6 (Item 2 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Acrobatics Motion Simulation based-on Dynamics. Interactive Content "Aerial Circus".

9/TI/7 (Item 3 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Dynamics-Based Interactive Acrobatic Motion Generator.

9/TI/8 (Item 4 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Motion Control of a Trampoline Gymnast Robot. Hopping Motion Control with a Consideration of Friction Disturbance.

9/TI/9 (Item 5 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Motion Control of a Trampoline Gymnast Robot. Control of the Continuous Hopping Motion.

9/TI/10 (Item 6 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Motion Control of a Trampoline Gymnast Robot. Computer simulations and Experiments.

9/TI/11 (Item 7 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Development of a trampoline gymnast robot.

9/TI/12 (Item 8 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Control of a Trampoline Robot. The fourth Report. Control of horizontal momentum.

9/TI/13 (Item 9 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Control of a Trampoline Robot. The third Report: Control of Continuous Straight Bounce.

9/TI/14 (Item 10 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Control of Hopping Robots on the Trampoline . 2nd Report. Computer Simulation of 3- link Model.

9/TI/15 (Item 11 from file: 94)
DIALOG(R)File 94:(c)2004 Japan Science and Tech Corp(JST). All rts. reserv.

Control of Hopping Machine. From Touchdown to Take off.

9/TI/16 (Item 1 from file: 6)
DIALOG(R)File 6:(c) 2004 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

Components, Assembly , and Use of a Trampoline . (ASTM Standard)

9/TI/17 (Item 2 from file: 6)
DIALOG(R)File 6:(c) 2004 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

A Refined Model Formulation for Static and Dynamic Analysis of Offshore Towers (Including the User's Guide for: TOJO 80 Tower-Joint Computer Program)

9/TI/18 (Item 1 from file: 95)
DIALOG(R)File 95:(c) 2004 FIZ TECHNIK. All rts. reserv.

Fracture and fatigue analysis of a hot tap tee
(Bruch- und Ermuedungsanalyse eines Warmanzapf-T-Verbindungstueckes)

16/5/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

14078 INSPEC Abstract Number: C9602-6150C-009
Title: **Giving** linkers **their due**
Author(s): **Szewerenko, L.** ; Mordoh, A.
Journal: Electronic Design vol.43, no.24 p.131-2, 134-5, 139-40
Publisher: Penton Publishing,
Publication Date: 20 Nov. 1995 Country of Publication: USA
CODEN: ELODAW ISSN: 0013-4872
SICI: 0013-4872(19951120)43:24L.131:GLT;1-Z
Material Identity Number: E140-95025
U.S. Copyright Clearance Center Code: 0013-4872/95/$2.00+1.00
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: **Linkers** are often maligned by software developers. Given today's hardware architectures, **linkers** sometimes appear to be the unnecessary remnants of an earlier archaic technology. Especially in the world of "self-hosted" **compilers** , **linkers** are often treated as the toolset's appendix. Yet we would argue that a wide range of post-**compilation** issues are best addressed by a sophisticated **linker** . Address computations, elimination of dead code, assignment of tasks to processors, and integration of multiple language environments are examples of work that should not be foisted on the **compiler** . Indeed, in the world of cross-development for real-time embedded systems, **linkers** are crucial to developing practical high-performance applications. A **linker** is the key to working with many issues central to embedded systems and digital signal processors (DSPs): use of RAM vs. ROM; separate code and data buses; shared vs. local memory; memory overlays; working with fixed hardware addresses; and effectively using both fast and slow memory. (0 Refs)
Subfile: C
Descriptors: program **compilers** ; programming environments; real-time systems; shared memory systems
Identifiers: **linkers** ; software development; address computations; post-**compilation** issues; multiple language environments; real-time embedded systems; digital signal processors; local memory; shared memory; memory overlays
Class Codes: C6150C (Compilers, interpreters and other processors); C6115 (Programming support)

16/5/2 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

5128905 INSPEC Abstract Number: C9601-6140D-030
Title: **C vs. Ada: arguing performance religion**
Author(s): **Syiek, D.**
Author Affiliation: Tartan Inc., Monroeville, PA, USA
Journal: Ada Letters vol.15, no.6 p.67-9
Publication Date: Nov.-Dec. 1995 Country of Publication: USA
CODEN: AALEE5 ISSN: 0736-721X
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: A discussion is given on the performance differences between C and Ada languages. It is concluded that: when written and **compiled** similarly, most Ada and C programs run equally efficiently; the quality of the **compiled** code is determined mostly by the quality of the **compiler** and not of the language; there are some cases where Ada code has an advantage; in C the burden of optimization is often in the hands of the programmer whereas In Ada, it is automated. It should come as no surprise that Ada provides optimization opportunities that C does not, and that it takes the burden of these optimizations off the back of the programmer. After all, the language was designed, from the beginning to permit high level programming of real time embedded systems. The differences are not enormous, but can be significant for applications requiring the most

...ation code possible. (0 Refs)
Subfile: C
Descriptors: Ada; C language; program **compilers** ; programming; real-time systems; software performance evaluation
Identifiers: Ada code; **compiler** ; performance religion; performance differences; C programs; optimization opportunities; high level programming ; real time embedded systems; efficient code
Class Codes: C6140D (High level languages); C0310F (Software development management); C6150C (Compilers, interpreters and other processors)

16/5/3 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

4695467 INSPEC Abstract Number: C9408-7150-002
Title: The implementation of asynchronous entry calls on two different architectures
Author(s): Fergany, A.; **Szewerenko, L.** ; Rabinowitz, M.; Solomon, E.N.; Pitarys, M.J.; Benjamin, C.L.
Author Affiliation: Tartan Inc., Monroeville, PA, USA
Part vol.1 p.486-95 vol.1
Publisher: IEEE, New York, NY, USA
Publication Date: 1993 Country of Publication: USA 2 vol. xvii+1171 pp.
ISBN: 0 7803 1295 3
U.S. Copyright Clearance Center Code: CH3306-8/93/0000-0486$1.00
Conference Title: Proceedings of NAECON '93 - National Aerospace and Electronics Conference
Conference Sponsor: IEEE
Conference Date: 24-28 May 1993 Conference Location: Dayton, OH, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Ada is required by the United States Air Force (USAF) for programming weapon system software. Each software development effort relies on an Ada Runtime System (RTS). The Common Ada Runtime System (CARTS) is composed of several Ada packages which are designed to furnish a standard interface between an Ada **compiler** and an Ada runtime system. CARTS also includes some packages which may be used directly by an application. This paper discusses implementations of the CARTS Asynchronous Calls package. In particular, it focuses on the components of the interface which have target specific properties, and it provides an analysis of the target and software costs for the asynchronous calls support. The cost associated with such support is represented by any overhead which must be accounted for in the application. (3 Refs)
Subfile: C
Descriptors: Ada; data structures; economics; military computing; program **compilers** ; programming environments; software packages; weapons
Identifiers: asynchronous entry calls; United States Air Force; weapon ...software; Ada Runtime System; Common Ada Runtime System; CARTS; Ada ...packages; standard interface; Ada **compiler** ; CARTS Asynchronous Calls ...package; software costs; overhead; data structures

18/5/1 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01455702 ORDER NO: AADAA-I9600776
THE DYNAMIC EXPANSION OF CLASS HIERARCHY (POLYMORPHISM)
Author: CLARK, CHARLES F.
Degree: PH.D.
Year: 1995
Corporate Source/Institution: THE UNIVERSITY OF UTAH (0240)
Source: VOLUME 56/09-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 4979. 88 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

The implementation of polymorphism in object-oriented programming languages provides a means by which new derived classes can be introduced into a running program. In statically typed, compiled object-oriented languages, such as C++, a program's class hierarchy is statically defined by a programmer in the source code for that program. After compilation this hierarchy of class definitions cannot be changed, or added to, **without modifying** and recompiling portions of the source **code** . In contrast, through the dynamic expansion of class hierarchy additions to this set of classes may be made while the program is running. This dynamic expansion enables a program to interact with objects of classes that are derived from known base classes, but whose class definitions were not known at the program's compilation time. For statically typed languages this expansion may be accomplished in a type-safe manner. The dynamic expansion of a program's class hierarchy is a powerful way of extending the reusability and extensibility features, offered by object-oriented languages, to running programs.

The increasing popularity of persistent object stores makes this an important capability. Through polymorphic load and store operations that read and write objects, along with their class information, support for persistent objects of dynamically derived types can be provided. Moreover, the repository of class definitions used to expand the class hierarchy of programs may change over time. The ability to evolve class definitions, substituting a new definition for an existing class implementation in the repository, provides support necessary for lifetime maintenance of programs.

This dissertation focuses on dynamic expansion of the class hierarchy of programs written in the C++ programming language. A system providing dynamic expansion, along with a supporting facility for storing persistent objects and class implementations, is described. The system uses a modified compiler and dynamic **linker** to accomplish the loading and storage of class implementations and objects. These capabilities are provided without making modifications or extensions to the C++ programming language. Our persistent object load and store operations fall short, however, in their support for operations on object graphs that contain certain types of pointers or unions. Both the evolution of class definitions and the demand-driven conversion of existing objects of evolved classes are supported by the system. Experimentation, which included modifying a discrete state simulation program to use our system, suggests that existing applications are easily adapted to make use of dynamic expansion.

18/5/2 (Item 1 from file: 94)
DIALOG(R)File 94:JICST-EPlus

5279433 JICST ACCESSION NUMBER: 03A0223063 FILE SEGMENT: JICST-E
Design Pattern Improvement by MixJuice Language.
TANAKA AKIRA (1); ICHISUGI YUJI (1)
(1) National Inst. Advanced Industrial Sci. and Technol., JPN
Joho Shori Gakkai Ronbunshi(Transactions of Information Processing Society of Japan), 2003, VOL.44,NO.SIG4 (PRO17), PAGE.25-46, FIG.26, TBL.2, REF.18
JOURNAL NUMBER: Z0778AAZ ISSN NO: 0387-5806

UNIVERSAL DECIMAL CLASSIFICATION: 681.3.02.001 681.3.06.004.14:800.92
LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan
DOCUMENT TYPE: Journal
ARTICLE TYPE: Original paper
MEDIA TYPE: Printed Publication
ABSTRACT: This paper presents benefits of the diffence based modules with design patterns. The design patterns are a catalog of designs to achive extensible architecture in object oriented languages. But the catalog also points out several problems which is caused by the limitation of the module system of the languages. The paper presents the problems can be fixed by MixJuice which is a Java-based language with the difference based modules. The difference based modules makes it possible to modify existing classes **without modifing** existing source **code** . The design patterns can be improved in 5 ways by the class modification. 1) An existing class can be participant to a design pattern by adding a interface. 2) Adding an abstract method in the superclass of existing class hierarchy makes a pattern more extensible. 3) Several concerns of a class can be hide thier information each other by implement the class by separated modules. 4) When run-time extensibility is not required, **link - time** extensibility by module selection can be used. It makes a system more type safe by avoiding useless downcast. 5) The **link - time** module selection also reduces number of classes and simplify a system. The paper presents layered class diagrams of design patterns for MixJuice and compare them to patterns for existing languages. (author abst.)

21/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06672744 E.I. No: EIP04017801945
Title: Dynamic compilation of C++ template code
Author: Cole, Martin J.; Parker, Steven G.
Corporate Source: Sci. Comp. and Imaging Institute School of Computing University of Utah, Salt Lake City, UT 84112, United States
Source: Scientific Programming v 11 n 4 SPEC. ISS. 2003. p 321-327
Publication Year: 2003
CODEN: SCIPEV ISSN: 1058-9244
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 0401W1
Abstract: Generic programming using the C++ template facility has been a successful method for creating high-performance, yet general algorithms for scientific computing and visualization. However, adding template code tends to require more template code in surrounding structures and algorithms to maintain generality. **Compiling** all possible expansions of these templates can lead to massive template bloat. Furthermore, compile-time binding of templates requires that all possible permutations be known at compile time, limiting the runtime extensibility of the generic code. We present a method for deferring the compilation of these templates until an exact type is needed. This dynamic compilation mechanism will produce the minimum amount of **compiled code** needed for a particular application, while maintaining the generality and performance that templates innately provide. Through a small amount of supporting code within each templated class, the proper templated **code** can be generated at runtime **without modifying** the **compiler** . We describe the implementation of this goal within the SCIRun dataflow system. SCIRun is freely available online for research purposes. 11 Refs.
Descriptors: Object oriented programming; Codes (symbols); Program **compilers** ; Natural sciences computing; Data structures; Subroutines; Computer operating systems; Data flow analysis; Function evaluation; Algorithms; Optimization
Identifiers: Generic programming; Software templates
Classification Codes:
723.1 (Computer Programming); 723.2 (Data Processing); 723.3 (Database Systems); 921.6 (Numerical Methods); 921.5 (Optimization Techniques)
723 (Computer Software, Data Handling & Applications); 722 (Computer Systems); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

21/5/2 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01317580 ORDER NO: AADMM-77278
THE ENTERPRISE CODE LIBRARIAN
Author: CHAN, ENOCH
Degree: M.SC.
Year: 1992
Corporate Source/Institution: UNIVERSITY OF ALBERTA (CANADA) (0351)
Source: VOLUME 31/04 of MASTERS ABSTRACTS.
PAGE 1828. 128 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984
ISBN: 0-315-77278-6

The parallelism of the Enterprise program is specified separately by attaching templates, called assets, to sequential modules through an Enterprise graph. The low-level communication code is generated by the system automatically according to the specified asset types. The separation of the specification of a program's source code and its parallel structure allows the program to be restructured easily **without changes** to the source **code** . This also allows a program to be adapted easily to the

changing resources available in a workstation environment.
This thesis presents the work on the design of the system architecture of Enterprise, the design and implementation of the Enterprise code librarian, and the Enterprise pre- **compiler** . The design of the architecture allows different components of the system to be implemented individually. Currently, several components have been implemented to allow Enterprise applications to be developed and tested. The Enterprise code librarian is designed for managing the source and **object code** of Enterprise applications. Since Enterprise applications are designed to run on a heterogeneous network of workstations, the code librarian takes this into account and provides a makefile generation utility to maintain multiple executable files for a variety of architectures. The Enterprise pre- **compiler** is used for converting sequential calls into remote procedure calls, changing return statements into reply statements, and substituting the function declarations with a suitable format to allow remote invocations. Several applications have been developed using the Enterprise environment. Experimental results show that Enterprise offers a cost effective and easy to learn method for the rapid construction of distributed software. (Abstract shortened by UMI.)

21/5/3 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

INSPEC Abstract Number: C9511-6110J-045
Title: SOM: Truly reusable class libraries in large programs
Author(s): Pennello, T.J.
Author Affiliation: MetaWare Inc., Santa Cruz, CA, USA
Conference Title: OOP '94/C++ World. Conference Proceedings p.131-3
Publisher: SIGS Publications, New York, NY, USA
Publication Date: 1994 Country of Publication: USA x+258 pp.
Conference Title: Proceedings of OOP'94/C++ World
Conference Date: 31 Jan.-4 Feb. 1994 Conference Location: Munich, Germany
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Because software construction is so expensive, the inflexibility of tight binding cannot be tolerated in large systems. IBM's System Object Model (SOM) was invented to solve this and other problems, lending the advantages of procedure libraries to OO technology. With SOM, objects are loosely coupled with their clients, providing binary independence. You can add methods to objects, insert classes in a class hierarchy, add or delete private data, generalize methods, and other things, all **without changing compiled** client **code** . The combination of DirectToSOM support for C++, SOM's binary independence, and SOM's ability to introduce OO programming in procedural languages gives real hope to the promise of truly reusable code. The guiding principle of SOM is: if the client doesn't have to edit his source code, he doesn't have to re-compile. (0 Refs)
Subfile: C
Descriptors: file organisation; object-oriented programming; program **compilers** ; software libraries; software reusability
Identifiers: SOM; reusable class libraries; large programs; software construction; tight binding; IBM System Object Model; OO technology; procedure libraries; binary independence; DirectToSOM; C++
Class Codes: C6110J (Object-oriented programming); C6110B (Software engineering techniques); C6115 (Programming support); C6120 (File organisation); C6150C (Compilers, interpreters and other processors)

21/5/4 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

4682049 INSPEC Abstract Number: C9407-6160J-008
Title: Texas: an efficient, portable persistent store

Author(s): Singhal, V.; Kakkad, S.V.; Wilson, P.R.
Author Affiliation: Dept. of Comput. Sci., Texas Univ., Austin, TX, USA
p.11-33
Editor(s): Albano, A.; Morrison, R.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 1993 Country of Publication: West Germany ix+446 pp.
ISBN: 3 540 19800 8
Conference Title: Proceedings of the Fifth International Workshop on Persistent Object Systems
Conference Date: 1-4 Sept. 1992 Conference Location: Pisa, Italy
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Texas is a persistent storage system for C++. A key component is the use of pointer swizzling at page fault time, which exploits existing virtual memory features to implement large address spaces efficiently on stock hardware, with little or no change to existing **compilers** . Long pointers are used to implement an enormous address space, but are transparently converted to the hardware-supported pointer format when pages are loaded into virtual memory. Runtime type descriptors and slightly modified heap allocation routines support pagewise pointer swizzling by allowing objects and their pointer fields to be identified within pages. If **compiler** support for runtime type identification is not available, a simple preprocessor can be used to generate type descriptors. This address translation employs the operating systems' existing virtual memories for caching, and a simple and flexible log-structured storage manager to improve checkpointing performance. Pagewise virtual memory protections are also used to detect writes for logging purposes, **without** requiring any **changes** to **compiled** **code** . This may degrade checkpointing performance for small transactions with poor locality of writes, but page differing and sub-page logging promise to keep performance competitive with finer-grained checkpointing schemes. Texas presents a simple programming interface; an application creates persistent objects by simply allocating them on the persistent heap. The implementation is relatively small, and is easy to incorporate into existing applications. The log-structured storage module easily supports advanced extensions. (30 Refs)
Subfile: C
Descriptors: data structures; object-oriented databases; program **compilers** ; software portability; storage allocation; virtual storage
Identifiers: Texas; portable persistent store; C++; programming interface ; pagewise pointer swizzling; page fault time; virtual memory; large address spaces; **compilers** ; long pointers; hardware-supported pointer format; runtime type descriptors; heap allocation routines; preprocessor; address translation; operating systems; data caching; log-structured storage manager; checkpointing performance; sub-page logging; write locality; page differing
Class Codes: C6160J (Object-oriented databases); C6120 (File organisation); C6110B (Software engineering techniques)

21/5/5 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

03288239 INSPEC Abstract Number: C89008339
Title: Hardware/software monitor for high level language program debugging
Author(s): Battistini, G.; Bissolotti, L.; Danese, G.; Dotti, D.; Franchi, G.
Author Affiliation: Dipartimento di Inf. e Sistemistica, Pavia Univ., Italy
Conference Title: Mini and Microcomputers and their Applications p. 75-8
Editor(s): Luque, E.
Publisher: Univ. Autonoma Barcelona, Barcelona, Spain
Publication Date: 1988 Country of Publication: Spain vi+680 pp.
ISBN: 84 7488 121 8
Conference Sponsor: Int. Soc. Mini & Microcomput
Conference Date: 27-30 June 1988 Conference Location: Sant Feliu de

Guixols, Spain
Availability: ISMM (Canadian Secretariat), Calgary, Alta., Canada
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: A hardware device able to monitor the execution of high level language programs has been developed, in order to support debugging operations. As a first implementation, an IBM Personal Computer programmed in the C language has been selected as the working environment. This tool allows the user to check the execution of the source program **instructions** that he has previously selected, **without modifying** the current process dynamics. A **compiler** , which produces an **object code** where each source statement is translated into a particular sequence of machine codes, has been used. Break-points may be associated with up to four source statements; then single instruction stepping is allowed. (5 Refs)
Subfile: C
Descriptors: high level languages; microcomputer applications; program debugging; supervisory programs
Identifiers: hardware/software monitor; break points; program execution monitoring; high level language program debugging; IBM Personal Computer; C language; source program instructions; process dynamics; **compiler** ; **object code** ; single instruction stepping
Class Codes: C6150G (Diagnostic, testing, debugging and evaluating systems)

21/5/6 **(Item 4 from file: 2)**
DIALOG(R)File 2:INSPEC
(c) 2004 Institution of Electrical Engineers. All rts. reserv.

02830891 INSPEC Abstract Number: C87017575
Title: Soft connections (interpretative interfaces for data communications)
Author(s): Wadsworth, B.
Author Affiliation: Austec Ltd., London, UK
Journal: Systems International vol.14, no.12 p.83-4
Publication Date: Dec. 1986 Country of Publication: UK
CODEN: SYIND8 ISSN: 0309-1171
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: It has been an aim of computer users for decades to achieve an open system where programs can run on disparate hardware **without change** , ideally at the **object code** level rather than at source code level. Ideally it should be possible to freely transfer data across different operating systems without change and for the systems to share interactively. Both the problems of data representation and the ability to move **object code** may be solved by using interpretative methods in a language **compiler** such as COBOL. An interface runs interpretative **object code** . It can also provide the option of reverse **compiling** into C to generate machine code for a particular processor. In a network made up of personal computers, minicomputers and mainframes, as is often found in a large company, an interpretative interface can take a file from one machine and load into another without altering the meaning of the data. (0 Refs)
Subfile: C
Descriptors: data communication systems; operating systems (computers); program **compilers** ; utility programs
Identifiers: open system; **object code** level; source code level; operating systems; data representation; interpretative methods; language **compiler** ; COBOL; reverse **compiling** ; machine code; personal computers; minicomputers; mainframes; interpretative interface
Class Codes: C5600 (Data communication equipment and techniques); C6150C (Compilers, interpreters and other processors); C6150E (General utility programs)

21/5/7 **(Item 1 from file: 6)**
DIALOG(R)File 6:NTIS
(c) 2004 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

1378899 NTIS Accession Number: AD-A193 297/9

Poker on the Cosmic Cube: The First Retargetable Parallel Programming Language and Environment

(Technical rept)

Snyder, L. ; Socha, D.
Washington Univ., Seattle. Dept. of Computer Science.

Corp. Source Codes: 005042231; 395224
Report No.: TR-86-02-05
Jun 86 17p
Languages: English
Journal Announcement: GRAI8818
Sponsored in part by Contract N00014-85-K-0328.

NTIS Prices: PC A03/MF A01
Country of Publication: United States
Contract No.: N00014-86-K-0264; NSF-DCR84-16878

This paper describes a technique for retargetting Poker, the first [illegible] parallel programming environment, to new parallel architectures. The specifics are illustrated by describing the retarget of Poker to CalTech's Cosmic Cube. Poker requires only three features from the target architecture: MIMD operation, message passing inter-process communication, and a sequential language (e.g. C) for the processor elements. In return Poker gives the new architecture a complete parallel programming environment which will compile Poker parallel programs **without modification** , into efficient **object code** for the new architecture.

Descriptors: High level languages; *Computer architecture; Coding; Computer programming; Computer programs; Efficiency; Environments; Parallel orientation; Sequences; **Compilers**

```
File 275:Gale Group Computer DB(TM) 1983-2004/Jul 13
         (c) 2004 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Jul 08
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jul 12
         (c) 2004 The Gale Group
File  16:Gale Group PROMT(R) 1990-2004/Jul 12
         (c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2004/Jul 09
         (c)2004 The Gale Group
File 624:McGraw-Hill Publications 1985-2004/Jul 15
         (c) 2004 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2004/Jul 15
         (c) 2004 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2004/Jul W1
         (c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Jun W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Jul 14
         (c) 2004 The Dialog Corp.
File 369:New Scientist 1994-2004/Jul W1
         (c) 2004 Reed Business Information Ltd.

Set     Items   Description
S1       2090   TRAMPOLINE? ? OR (BRANCH OR IMPORT)()STUB? ?
S2      99269   (TARGET? ? OR DESTINATION OR BRANCH?? OR CALL??? OR ADDRESS
             OR ADDRESSES)(7N)(FAR OR DISTANT OR REMOT??)
S3      34938   (DISPLACE???? OR OFFSET OR OFF()SET)(7N)(BIG?? OR LARGE?? -
            OR MUCH OR GREAT???)
S4       8568   (UNCHANGED OR UNMODIFIED OR ("NOT" OR WITHOUT OR T)(2W)(CH-
            ANG??? OR MODIF???? OR MODIFICATION OR ALTER??? OR ALTERATION-
            ))(7N)(CODE? ? OR INSTRUCTION? ?)
S5    3118313   COMPIL? OR LINK??? OR ASSEMB???
S6          0   S1(100N)S4(100N)S5
S7         68   S1(100N)S5
S8          0   S1(100N)S2:S3(100N)S5
S9         51   RD S7 (unique items)
S10        32   S9 NOT PD>19991008
S11       145   STUB()(CODE? ? OR ROUTINE? ?)
S12         2   (S1 OR S11)(100N)S4
S13        33   (S1 OR S11)(100N)S2:S3
S14        22   RD (unique items)
S15        20   S14 NOT (PD>19991008 OR S10 OR S12)
S16      9903   (COMPILED OR OBJECT)(1W)CODE
S17        60   S16(10N)S4
S18        38   S17(100N)(S2:S3 OR S5)
S19        24   RD (unique items)
S20        20   S19 NOT (S10 OR S12 OR S15 OR PD>19991008)
```

15/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02296333 SUPPLIER NUMBER: 54622881 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Making sense of the COM-CORBA debate. (Technology Information)
yee, andre
UNIX Review's Performance Computing, 17, 6, D3(1)
June, 1999
LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 2900 LINE COUNT: 00240

... the client's perspective, a method call is invoked through interface pointers running in process. When a client makes a **remote call**, the **call** is actually handled by a client-side **stub code** known as a proxy. The proxy packs the call parameters into a message and delivers the message to the server...

15/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02207494 SUPPLIER NUMBER: 20964144 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Understanding Interface Definition Language: a developer's survival guide. (Technology Tutorial)
Hludzinski, Bill
Microsoft Systems Journal, v13, n8, p51(12)
August, 1998
ISSN: 0889-9932 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 7206 LINE COUNT: 00642

... Review
IDL was originally part of the Open Software Foundation's Distributed Computing Environment (DCE). It described function interfaces for **Remote** Procedure **Calls** (RPCs), so that a compiler could generate proxy and **stub code** that marshaled parameters between machines. MIDL is Microsoft's IDL compiler. In addition, Microsoft developed its own Object Definition Language...

15/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02057187 SUPPLIER NUMBER: 19272199 (USE FORMAT 7 OR 9 FOR FULL TEXT)
ActiveX/COM. (Technology Tutorial)
Box, Don
Microsoft Systems Journal, v12, n5, p95(9)
May, 1997
ISSN: 0889-9932 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 4531 LINE COUNT: 00381

... s process for the duration of the UseThisObject method call. This is required to allow the inproc object to receive **calls** from the **remote** object. If the **remote** object does not AddRef the proxy it receives as the (in) parameter to UseThisObject, all is well because the **stub routine** for UseThisObject will automatically release the proxy to pbar, tearing down the Bar stub once the method call has completed...

15/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01980810 SUPPLIER NUMBER: 18686178 (USE FORMAT 7 OR 9 FOR FULL TEXT)
OPEN GROUP UNBUNDLES DISTRIBUTED COMPUTING ENVIRONMENT SYSTEM TO ENABLE JAVA TO ACCESS NETWORKED DATABASES.

Computergram International, n991, pCGN09040018
Sep 4, 1996
ISSN: 0268-716X LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 448 LINE COUNT: 00040

TEXT:
...and the end of next year the Institute intends to modify the DCE Interface Definition Language compiler to generate Java **stub code** and provide access to the DCE application programming interface through a set of Java classes modelled on Hewlett-Packard Co...

...may be called C++ for DCE. Deliverables will include Distributed Computing Environment Interface Definition Language to Java mapping specification, abstract **Remote** Procedure **Call** class specification; DCE Interface Definition Language to Java compiler and Distributed Computing Environment run-time encapsulated as Java native methods.

15/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01957786 SUPPLIER NUMBER: 18169229 (USE FORMAT 7 OR 9 FOR FULL TEXT)
How OLE and COM solve the problems of component software design. (component object model) (Technology Tutorial)(Tutorial)
Brockschmidt, Kraig
Microsoft Systems Journal, v11, n5, p63(15)
May, 1996
DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 12758 LINE COUNT: 01020

... can then clean up any reference counts to the object on behalf of the missing client.
The necessary proxy and **stub code** for most standard interfaces (those defined by Microsoft) is built into the system. If you define your own custom interface, you have to supply your own proxy/ **stub code** . Fortunately, the MIDL compiler will generate this code for you from an IDL file. Compile the code into a DLL...

...not necessary to have an OLE implementation on the server machine to interoperate with COM on the client machine. All **remote** interface **calls** are transmitted with DCE-compatible RPC so any DCE-aware system receives the calls and converts them to fit any...

15/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01682865 SUPPLIER NUMBER: 15386456 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Can DCE fulfill its promise? Lack of development tools and OS compatibility hinder corporate acceptance. (distributed computing environment) (PC Week LABS: Tech View) (PC Week Special Report: Client/Server)
Gallagher, Bob
PC Week, v11, n20, p105(1)
May 23, 1994
ISSN: 0740-1604 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1190 LINE COUNT: 00095

... migrate to the PC.
Following is a look at the nuts and bolts of this potentially powerful client/server technology.
Remote calls
RPC is the cornerstone of distributed computing; it enables an application to execute a distributed component on another machine simply by sending a message addressed to the remote program.
Building RPC-capable programs requires the insertion of **stub code**

... handle the communication between the main program and the remote procedure. **Stub code** is usually complex and requires a high-level network programmer to create it, but products like NobleNet Inc.'s WinRPC ...

...RPC (see PC Week, Dec. 13, 1993, Page 89) can greatly simplify the creation of RPC-capable applications.

After the **stub code** is in place, it uses the RPC run-time library routines to determine which transport layer should be used. It...

15/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01674724 SUPPLIER NUMBER: 15061427 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Amid praise and catcalls, DCE comes into the open. (the Open Software Foundation's Distributed Computing Environment)
Vaughan, Jack
Software Magazine, v14, n3, p55(5)
March, 1994
ISSN: 0897-8085 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3600 LINE COUNT: 00292

... to Sun Unix boxes. And Sun's NFS supports Kerberos security services.

Be it Sun's or OSF's, the **remote** procedure **call** is not strange to programmers, being similar to a sub-routine call. Still, only a relative handful of programmers have experience writing software for communications, or with organizing program threads that are initiated while **remote calls** are being completed. With the RPC, a calling program initiates client **stub code** that actually performs communications. As a result, individual procedures in an application can run on any supported computer connected to ...

15/3,K/8 (Item 8 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01663746 SUPPLIER NUMBER: 14822291 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Writing DCE programs. (how to write a Distributed Computing Environment application) (Tutorial)
Shirley, John
UNIX Review, v12, n1, p35(10)
Jan, 1994
DOCUMENT TYPE: Tutorial ISSN: 0742-3136 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 4434 LINE COUNT: 00365

...ABSTRACT: to write a Distributed Computing Environment (DCE) application is described. The simplest way to implement client-server applications is the **remote** procedure **call** (RPC) mechanism, which keeps the details of network communications out of the application code. An RPC in client application code looks like a local procedure call because it is actually a call to client **stub code** . Stubs are surrogate code that supports RPCs: the client stub uses the RPC run-time library to communicate with the...

...the remote procedure in the server application code. The code executes in the server's address space, and the server **stub code** communicates output to the client stub code using the RPC run-time library when the remote procedure has been executed...
... differences between regular and client-server programs were discussed. This month, the construction of a DCE application is presented.

The **remote** procedure call (RPC) memchanism is the simplest way to implement client-server applications, because it keeps the details of network...

...between your application code and the RPC mechanism during a remote procedure call. In client application code, a remote procedure **call** look like a local procedure **call** , because it is actually a call to client **stub code** .

A stub is surrogate code that supports **remote** procedure **calls** . Later on we'll discuss how stubs are created and how they work. The client stub communicates with the server...

...of standard run-time routines that support all DCE RPC applications.

The server's RPC- run-time library receives the **remote** procedure **call** and communicates client information to the server stub. The server **stub code** invokes the remote procedure in the server application code, which executes in the server's **address** space. When the server finishes executing the **remote** procedure, the server stub using the RPC run-tEme library. Finally, the client **stub code** returns to the client's application code.

Client and server developed of an application can occur in parallel and on...

...uses special hardware, such as an array processor. In our example, the client performs an arithmetic operatioan on arrays by **calling** a **remote** procedure that uses the array processor. The remote procedure executives on the server system, taking two arrays as argyments and...in client and server application code.

* A client stub file linked with the client portion of the application. During a **remote** procedure **call** , the client **stub code** is intermediate between your client appliocation code and the RPC run-time library.

* A server stub file linked with the server portion of the application. During a **remote** procedure **call** , the server **stub code** is intermediate between your server application code and the RPC run-time library.

The IDL compiler goes through two phases...

15/3,K/9 **(Item 9 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 1994 The Gale Group. All rts. reserv.

00619642 SUPPLIER NUMBER: 14807663 (USE FORMAT 7 OR 9 FOR FULL TEXT)
NobelNet's WinRPC slashes development time, expenses. (remote procedure calls) (Software Review) (PC Week LABS: First Looks) (includes related article on NobelNet's WinRPC 1.0) (Evaluation)
Gallagher, Bob
PC Week, v10, n49, p89(1)
Dec 13, 1993
DOCUMENT TYPE: Evaluation ISSN: 0740-1604 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 929 LINE COUNT: 00074

... fairly simple (see graphic, Page 101). Local procedure calls are replaced by an RPC procedure, or stub, that routes the **call** to a **remote** procedure with the same name -- but on another machine.

The **stub routine** in the client program packs together arguments, such as the name of the routine and instructions on how the routine...

...a type recognizable on the server platform; when the server routine receives it, the server passes it to the receiving **stub routine** , which unpacks the data, and the remote procedure is executed.

If the server program needs to return data to the...

15/3,K/10 **(Item 10 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 1994 The Gale Group. All rts. reserv.

00619642 SUPPLIER NUMBER: 14357968 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Cool: system support for distributed programming. (one of eight articles on concurrent object-oriented programming; special issue) (Technical)
Lea, Rodger; Jacquermot, Christian; Pillevesse, Eric
Communications of the ACM, v36, n9, p37(10)
Sep, 1993
DOCUMENT TYPE: Technical ISSN: 0001-0782 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 7370 LINE COUNT: 00594

... used to trap the normal-function invocation and replace it by a remote invocation which marshals the parameters, issues a **remote** procedure **call** , and unmarshals the results (Figure 3). At the receiver, a dispatch procedure, which is part of the up-call function...

...model to the generic run-time model. This may be achieved through the use of preprocessors to generate the correct **stub code** to access the GRT functionality and the use of an up-call table to allow the GRT to access language...

15/3,K/11 (Item 11 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01517717 SUPPLIER NUMBER: 12182948 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Go Forth and multiply. (distributed architectures)
McLachlan, Gordon
LAN Computing, v3, n5, p19(3)
May, 1992
ISSN: 1055-1808 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2421 LINE COUNT: 00184

... be exported;
A network programmer writes the interface specifications in the IDL and compiles it to get client and server **stub code** ;
And real client programs and server procedures are written using the generated header files.
To run the client-server system:
The server process daemon is executed and registers itself with the network, specifying the procedures it is providing to **remote** clients;
The client application **calls** a **remote** procedure through the local stub. The stub retrieves the network address of the server process daemon from the network directory service. The **stub code** then sends a message to the server daemon identifying the server procedure being called, including the parameters and security information...

15/3,K/12 (Item 12 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01386272 SUPPLIER NUMBER: 09716367 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Experiences with the Amoeba distributed operating system. (technical)
Tanenbaum, Andrew S.; Renesse, Robbert van; Staveren, Hans van; Sharp, Gregory J.; Mullender, Sape J.; Jansen, Jack; Rossum, Guido van
Communications of the ACM, v33, n12, p46(18)
Dec, 1990
DOCUMENT TYPE: technical ISSN: 0001-0782 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 14734 LINE COUNT: 01130

... message that unblocks the client. The combination of sending a request message, blocking, and accepting a reply message forms the **remote** procedure **call** , which can be encapsulated using **stub routines** , to make the entire **remote** operation look like a local procedure **call** . (For other possibilities see [28]).
The structure of a capability is shown in Figure 2. It is 128 bits long...

15/3,K/13 (Item 13 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01337008 SUPPLIER NUMBER: 08843334
Retargetable stub generator for a remote procedure call facility. (technical)
Tham, Y.K.; Bhonsle, S.K.
Computer Communications, v13, n6, p323(8)
July-August, 1990
DOCUMENT TYPE: technical ISSN: 0140-3664 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

ABSTRACT: A **remote** procedure **call** system to support **calls** from Lisp clients to C servers is extended by defining a language to specify how **stub code** is to be generated for a target language, and implementing a language-independent code generator to execute the stub generation...

15/3,K/14 (Item 14 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01319448 SUPPLIER NUMBER: 07928710 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Climbing up to Windows. (Microsoft Corp.'s Windows Software Development Kit)
Davidson, Mark
Computer Language, v6, n11, p91(9)
Nov, 1989
ISSN: 0749-2839 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 4680 LINE COUNT: 00353

... well as a collection of assembly language macros. However, almost all of the functions in the supplied libraries use the **far** pascal **calling** convention since the pieces of Windows you call may not be in your code space. In fact, they could be anywhere in memory. Windows is constantly moving or discarding code and data to make room. Thus, the libraries are really **stub routines** that pass control to a supervisory portion of Windows. This portion of Windows is responsible for transferring control to the...

15/3,K/15 (Item 15 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01268098 SUPPLIER NUMBER: 07895540
RPC tools pave way for cooperative processing. (remote procedure call)
Kobielus, James
Network World, v6, n46, p1(5)
Nov 20, 1989
ISSN: 0887-7661 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: **Remote** procedure **call** (RPC) application development tools are being developed that will help software developers build distributed network applications by automating the process...

...to dispersed, networked computers. RPC tools divide a source-code program into client and server pieces, then generate automatically the **stub code** .

15/3,K/16 (Item 1 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

03231821 Supplier Number: 46626632 (USE FORMAT 7 FOR FULLTEXT)
THE NEWLY-NAMED OPEN GROUP RESEARCH INSTITUTE OFFERS DISTRIBUTED COMPUTING ENVIROMENT WEB TECHNOLOGY
Computergram International, n2976, pN/A
August 13, 1996
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 619

... quarter and the end of next year the Institute will modify the DCE Interface Definition Language compiler to generate Java **stub code** and provide access to the Distributed Computing Environment application programming interface through a set of Java classes modelled on Hewlett...

...easily be called C++ for DCE. Deliverables will include a DCE Interface Definition Language to Java mapping specification, an abstract **Remote** Procedure **Call** class specification; a DCE Interface Definition Language to Java compiler; and a DCE run-time environment encapsulated as Java native...

15/3,K/17 **(Item 1 from file: 160)**
DIALOG(R)File 160:Gale Group PROMT(R)

02357654
RPC Tools pave way for cooperative processing
Network World November 20, 1989 p. 1,40+

Remote procedure **call** (RPC) application development products are being used more often to help build network applications for different environments, according to J...

... Banyan Systems and Sun Microsystems allow software engineers to divide applications programs and redistribute them among separate machines. The RPC **stub code** (which manages communications between the distributed procedures) and the linked communications protocol stacks handle the client-server network link. RPC...

15/3,K/18 **(Item 1 from file: 148)**
DIALOG(R)File 148:Gale Group Trade & Industry DB

09658681 SUPPLIER NUMBER: 19433262 (USE FORMAT 7 OR 9 FOR FULL TEXT)
App development software: what do TI C80-based image processing developers really want?
Robinson, Laura
Advanced Imaging, v12, n2, p45(3)
Feb, 1997
ISSN: 1042-0711 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 2097 LINE COUNT: 00171

... to work together; MIL commands make calls to the Native Library commands.

Basically, the Native Library is a set of **stub routines** , one for each supported opcode. Native Library commands are initiated by the host and make **remote** procedure **calls** to the actual processing functions on the C80 (the Native Library Shell resides in the on-board SDRAM processing memory...

15/3,K/19 **(Item 1 from file: 624)**
DIALOG(R)File 624:McGraw-Hill Publications

0639296
A DCE WORTH WAITING FOR

Open Computing February, 1995; Pg 43; Vol. 12, No. 2
Journal Code: UNIX ISSN: 0739-5922
Section Heading: COVER STORY
Word Count: 581 *Full text available in Formats 5, 7 and 9*

BYLINE:
RIKKI KIRZNER

TEXT:
...messages generated by different systems.

Improved security: A generic security service has been added to allow systems not based on **remote** procedure **call** (RPC) (such as message-passing applications) to exploit DCE security. Extended registry attributes enable users to log onto one machine...

...data from one language to another.

Performance: OSF improved its interface definition language (IDL) compiler so it generates smaller, cleaner **stub code** and supports new IDL constructs. (**Stub code** connects a client making an RPC **call** with a **remote** server.) There are also new RPC enhancements and optimizations.

Version 1.1 also includes a gateway that allows network file...

15/3,K/20 **(Item 1 from file: 15)**
DIALOG(R)File 15:ABI/Inform(R)

01631119 02-82108
Modern languages and Microsoft's component object model: Programming COM made simple
Gray, David N; Hotchkiss, John; LaForge, Seth; Shalit, Andrew; Weinberg, Toby
Communications of the ACM v41n5 PP: 55-65 May 1998
ISSN: 0001-0782 JRNL CODE: ACM
WORD COUNT: 4623

...TEXT: adds slots for supporting a reference count and aggregation of interfaces. The elements of the v-table are automatically generated **trampoline** functions that transform their arguments into Dylan objects and call a corresponding Dylan generic function. The Dylan <IUnknown> class also... QueryInterface, AddRef, and Release. These same methods can be accessed either directly through normal Dylan method dispatch, or by a **remote** client **calling** through the v-table **trampolines** . This dual implementation is transparent to the clients.

File 347:JAPIO Nov 1976-2004/Mar(Updated 040708)
(c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200444
(c) 2004 Thomson Derwent
File 348:EUROPEAN PATENTS 1978-2004/Jul W01
(c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20040701,UT=20040624
(c) 2004 WIPO/Univentio

Set	Items	Description
S1	22	AU=(SZEWERENKO L? OR SYIEK D? OR CYRAN R? OR CYRAN B?)
S2	1	S1 AND (TRAMPOLINE? ? OR (BRANCH OR IMPORT)()STUB? ? OR ST-UB()(ROUTINE? ? OR CODE? ?))

2/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01513858
Linking of applications into devices having overlays and shadow memories
Bindung von Anwendungen in Geraten mit Uberlagerung und Schattenspeichern
Liaison d'applications dans des appareils aux memoires d'ombre et a la superposition

PATENT ASSIGNEE:
Texas Instruments Incorporated, (279078), 7839 Churchill Way, Mail Station 3999, Dallas, Texas 75251, (US), (Applicant designated States: all)
INVENTOR:
CYRAN, Robert J. , 320 Links Court, 15626, Delmont, (US)
SYIEK, David A. , 103 Walnut Ridge Drive, 15238-1210, Pittsburgh, (US
LEGAL REPRESENTATIVE:
Holt, Michael et al (50422), Texas Instruments Ltd., EPD MS/13, 800 Pavilion Drive, Northampton Business Park, Northampton NN4 7YL, (GB)
PATENT (CC, No, Kind, Date): EP 1265136 A2 021211 (Basic)
APPLICATION (CC, No, Date): EP 2002100679 020607;
PRIORITY (CC, No, Date): US 296443 P 010608
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE; TR
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/445
ABSTRACT WORD COUNT: 121
NOTE:
Figure number on first page: 3

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200250	443
SPEC A	(English)	200250	3563
Total word count - document A			4006
Total word count - document B			0
Total word count - documents A + B			4006

INVENTOR:
CYRAN, Robert J ...

...US)
SYIEK, David A ...

...SPECIFICATION make visible in the address space. The routine performs the operation (Step 105). When the call returns back to the **stub code** , the stub performs whatever action is required to recover the memory map to the state it was prior to the call (Step 106) and then returns to the user code. The informationnecessary to build **stub routines** is attached to the existing memory used by the linker 35. For each memory device that is present in the...

Search Results

Search Results for: **[(unchanged or unmodified or ("not" or without or t) <NEAR/2> (chang* or modif* or alter*)) <NEAR/7> (code* or instruction*) <AND>(((trampoline* or branch stub* or import stub* or stub routine* or stub code*)))]**
Found **11** of **139,567 searched.**

Search within Results

> Advanced Search

> Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 11 of 11 **short listing**

1 Dynamic Adaptive compilation: An infrastructure for adaptive dynamic optimization 100%

Derek Bruening , Timothy Garnett , Saman Amarasinghe
Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization March 2003

Dynamic optimization is emerging as a promising approach to overcome many of the obstacles of traditional static compilation. But while there are a number of compiler infrastructures for developing static optimizations, there are very few for developing dynamic optimizations. We present a framework for implementing dynamic analyses and optimizations. We provide an interface for building external modules, or clients, for the DynamoRIO dynamic code modification system. This interface abstracts awa ...

2 Dynamic translation: Retargetable and reconfigurable software dynamic translation 99%

K. Scott , N. Kumar , S. Velusamy , B. Childers , J. W. Davidson , M. L. Soffa
Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization March 2003

Software dynamic translation (SDT) is a technology that permits the modification of an executing program's instructions. In recent years, SDT has received increased attention, from both industry and academia, as a feasible and effective approach to solving a variety of significant problems. Despite this increased attention, the task of initiating a new project in software dynamic translation remains a difficult one. To address this concern, and in particular, to promote the adoption of SDT techn ...

3 Motif/Lesstif Application Development: A tutorial designed to help you build your own GUI 97%

Glen Wiley
Linux Journal August 1999

4 Advanced control flow in Java card programming 96%

Peng Li , Steve Zdancewic
ACM SIGPLAN Notices , Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools June 2004
Volume 39 Issue 7

Java Card technology simplifies the development of smart card applications by providing a high-level programming language similar to Java. However, the master-slave programming model used in current Java Card platform creates control flow difficulties when writing complex card programs, making it inconvenient, tedious, and error-prone to implement Java Card applications. This paper examines these drawbacks of the master-slave model and proposes a concurrent thread model for developing future Jav ...

5 Compatible genericity with run-time types for the Java programming language 96%

Robert Cartwright , Guy L. Steele
ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications October 1998
Volume 33 Issue 10

The most serious impediment to writing substantial programs in the Java™ programming language is the lack of a *gentricity* mechanism for abstracting classes and methods with respect to type. During the past two years, several research groups have developed Java extensions that support various forms of genericity, but none has succeeded in accommodating general type parameterization (akin to Java arrays) while retaining compatibility with the existing. Java Virtual Machine. In thi ...

6 Applications 2: Implementation and performance evaluation of CONFLEX-G: grid-enabled molecular conformational space search program with OmniRPC 77%

Yoshihiro Nakajima , Mitsuhisa Sato , Hitoshi Goto , Taisuke Boku , Daisuke Takahashi
Proceedings of the 18th annual international conference on Supercomputing June 2004

CONFLEX-G is the grid-enabled version of a molecular conformational space search program called CONFLEX. We have implemented CONFLEX-G using a grid RPC system called OmniRPC. In this paper, we report the performance of CONFLEX-G in a grid testbed of several geographically distributed PC clusters. In order to explore many conformation of large bio-molecules, CONFLEX-G generates trial structures of the molecules and allocates jobs to optimize a trial structure with a reliable molecular mechanics m ...

7 Extending Java for high-level Web service construction 77%

Aske Simon Christensen , Anders Møller , Michael I. Schwartzbach
ACM Transactions on Programming Languages and Systems (TOPLAS) November 2003
Volume 25 Issue 6

We incorporate innovations from the <bigwig> project into the Java language to provide high-level features for Web service programming. The resulting language, JWIG, contains an advanced session model and a flexible mechanism for dynamic construction of XML documents, in particular XHTML. To support program

development we provide a suite of program analyses that at compile time verify for a given program that no runtime errors can occur while building documents or receiving form input, and ...

8 Interconnecting heterogeneous computer systems 77%

David Notkin , Andrew P. Black , Edward D. Lazowska , Henry M. Levy , Jan Sanislo , John Zahorjan

Communications of the ACM March 1988

Volume 31 Issue 3

A software structure created by the Heterogeneous Computer Systems (HCS) Project at the University of Washington was designed to address the problems of heterogeneity that typically arise in research computing environments.

9 Optimising hot paths in a dynamic binary translator 77%

David Ung , Cristina Cifuentes

ACM SIGARCH Computer Architecture News March 2001

Volume 29 Issue 1

In dynamic binary translation, code is translated "on the fly" at run-time, while the user perceives ordinary execution of the program on the target machine. Code fragments that are frequently executed follow the same sequence of flow control over a period of time. These fragments form a hot path and are optimised to improve the overall performance of the program.Multiple hot paths may also exist in programs. A program may choose to execute in one hot path for some time, but later switch to anot ...

10 Mimic: a fast system/370 simulator 77%

C. May

ACM SIGPLAN Notices , Papers of the Symposium on Interpreters and interpretive techniques July 1987

Volume 22 Issue 7

Software simulation of one computer on another tends to be slow. Traditional simulators typically execute about 100 instructions on the host machine per instruction simulated. Newer simulators reduce the expansion factor to about 10, by saving and reusing translations of individual instructions. This paper describes an experimental simulator which takes the progression one step further, translating groups of instructions as a unit. This approach, combined with flow analysis, reduces the expansio ...

11 Early experience with message-passing on the SHRIMP multicomputer 77%

Edward W. Felten , Richard D. Alpert , Angelos Bilas , Matthias A. Blumrich , Douglas W. Clark , Stefanos N. Damianakis , Cezary Dubnicki , Liviu Iftode , Kai Li

ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture May 1996

Volume 24 Issue 2

The SHRIMP multicomputer provides virtual memory-mapped communication (VMMC), which supports protected, user-level message passing, allows user programs to perform their own buffer management, and separates data transfers from control transfers so that a data transfer can be done without the intervention of the receiving node CPU. An important question is whether such a mechanism can indeed deliver all of the available hardware performance to applications which use conventional message-passing l ...